UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

PLAN 2008

SISTEMA DE GAMIFICACIÓN PARA LA GESTIÓN DE PROYECTOS SOFTWARE

Tutor: José Luis López Cuadrado

Autor: Alberto González Tajuelo

Agradecimientos

Me gustaría expresar mi agradecimiento a todas las personas que han aportado su granito de arena a mi vida. Gracias de corazón.

Quiero dar las gracias a mi padre Antonio por todo el apoyo que me ha brindado. Gracias a él mi calidad en la escritura ha mejorado mucho. Aunque parezca una tontería saber escribir te mejora la forma de expresarte y, lo que es mejor aún, que otras personas te entiendan.

Agradecer a Beatriz por el tiempo que ha dedicado y toda la ayuda que me ha prestado dando su punto de vista. También por estar en todos los momentos de mi vida, tanto buenos como malos.

Por último, me gustaría agradecer a mi tutor José Luis por ser un excelente profesor y hacer de la universidad Carlos III de Madrid un sitio más acogedor y más profesional.

Resumen

El objetivo es comprender el funcionamiento práctico de la creando un sistema de simulación de gamificación cercano a la realidad. Uno de los problemas cuando se realiza Gestión de Proyectos es que “el papel lo aguanta todo”. Y, cuando llega el momento de hacer del proyecto una realidad “todo se desploma”. Hay prisas, momentos de incertidumbre, el cliente no se comporta como era esperado, el equipo no alcanza las expectativas o suceden eventos no esperados.

Mediante un simulador se pueden, en cierta medida, disminuir los riesgos inherentes a un proyecto. Es una herramienta muy eficaz para poner en práctica los conocimientos adquiridos.

Contenido

Índice de figuras

Índice de tablas

Capítulo 1: Introducción

En este capítulo se da una visión general del Trabajo de Fin de Grado y los temas que van a ser desarrollados en el resto de capítulos.

1.1. Motivación

Para dedicar tiempo a un trabajo o un proyecto es importante tener un listado de factores que inciten a la realización del mismo. A continuación, muestro los que han sido las principales motivaciones que me han llevado a realizar este trabajo:

- Diseño de una aplicación web usando mis conocimientos para hacer que una idea se haga realidad. La diferencia entre una idea y un proyecto es simple. La idea es el descubrimiento de una necesidad mal cubierta y una solución con un producto o servicio con el que poder cubrirlo. El proyecto es la realidad de la idea. Es decir, el producto o servicio que se va a ofrecer para cubrir el problema detectado.

- Construir un prototipo de simulador en el que se puede dirigir un proyecto, ver qué problemas se tienen en el día a día y aprender de cómo hacer para poder mejorar.

- Implementar Integración Continua. Es decir, cuando se realice un cambio en la aplicación, automáticamente se descargue el código, se compile y se descarguen las dependencias, se hagan pruebas unitarias, de aceptación y funcionales. Y en caso de fuera todo correcto, se despliegue en el servidor y, si en algún caso, hubiera algún error, recibir un correo con el error producido.

- En un futuro, el desarrollo puede llevar a implantar un simulador en alguna asignatura que pueda ayudar a nuevos compañeros para que puedan mejorar su aprendizaje de asignaturas relacionadas con Dirección de Proyectos Software.

1.2. Objetivos

La meta principal es realizar el análisis, diseño e implementación de un simulador para poder dirigir y gestionar proyectos. Esta meta principal va a llevar consigo un listado de objetivos que serán cruciales para poder llevarlo a cabo. Son los siguientes:

- **Gamificación para mejora en el aprendizaje de estudiantes.** Ayudar a los estudiantes a que puedan aprender más y mejor sobre Dirección de Proyectos Software.

- **Uso de Laravel.** Un framework de desarrollo en PHP que implementa un modelo MVC (Modelo-Vista-Controlador) en el que se agiliza el tiempo de desarrollo de forma incremental.

- **Uso de Composer.** Composer es un gestor de dependencias de paquetes en PHP para facilitar cuando se trabaja con distintos proyectos (por ejemplo, Laravel es una dependencia del proyecto).

- **Uso de Bower.** Bower es un gestor de componentes para la parte web de una aplicación: HTML, CSS, JavaScript, fuentes e incluso imágenes. Es una herramienta necesaria cuando se quiere tener versionado los componentes que se usan.

- **Uso de Control de Versiones.** Se utilizará GIT para poder gestionar los cambios que se realicen la aplicación, poder volver atrás si existiera algún problema o similar.

- **Uso de Integración Continua.** Integración Continua significa que todo el trabajo de distintos desarrolladores es unido en cuanto es posible para poder compilarlo y ejecutar las pruebas unitarias (y poder recibir un reporte de si ha existido algún problema al unirlo). Jenkins [7] es una herramienta ideal para realizar esta tarea.

- **Uso de Entrega Continua y Despliegue Continuo.** Jenkins [7] puede ayudar a desplegar de forma continua la aplicación con los últimos cambios.

- **Diseño y gestión de una base de datos.** Se debe realizar una base de datos para poder almacenar los datos que sean necesarios en la aplicación. También sería necesario poder realizar migraciones para añadir los últimos cambios modificados de la base de datos.

- **Documentación del proyecto.** Realizar un estudio donde se analiza y se elabora todos los documentos necesarios del proyecto.

1.3. Estructura de la memoria

Para comprender mejor la estructura de la memoria, este apartado se ha dedicado a detallar cada uno de los capítulos:

Capítulo 1: Introducción. En este capítulo se da una visión general del Trabajo de Fin de Grado y del tema elegido. Se dan unas pinceladas básicas tanto de la dirección de proyectos software como del funcionamiento básico de un simulador. Por último, se presentan los objetivos y las motivaciones que han llevado a elegir este Trabajo de Fin de Grado.

Capítulo 2: Marco regulador. El marco regulador permite encuadrar el proyecto en todos los aspectos legales necesarios. Se valorarán los riesgos, responsabilidades profesionales, responsabilidades éticas, privacidad y seguridad, etc.

Capítulo 3: Entorno socio-económico. Hablar del entorno socio-económico es hablar de qué presupuesto sería necesario para construir el proyecto y valorar qué implicaciones sociales, económicas, medioambientales, éticas y su posible plan de explotación.

Capítulo 4: Contexto y Definición del Problema. Este capítulo trata sobre una explicación básica de la dirección de proyectos software y una breve explicación sobre los términos más usados. Además de una explicación del problema planteado.

Capítulo 5: Análisis del sistema. Se realiza un análisis del sistema después de conocer en qué se basa la Dirección de proyectos software y qué podría aportar de valor.

Capítulo 6: Diseño del sistema. En el diseño del sistema se realiza una visión más profunda y la interacción del usuario con el sistema. Se definen las clases del sistema, así como se especifica el entorno tecnológico.

Capítulo 7: Pruebas del sistema y manual de usuario. Para comprobar que el sistema funciona correctamente es necesario realizar pruebas. Estas pruebas añaden estabilidad al sistema y proporcionan una medida excepcional para corregir posibles errores.

Capítulo 8: Planificación del proyecto. Todo proyecto (como éste que se presenta como Trabajo de Fin de Grado) debe ser planificado durante un intervalo de tiempo. Se incluye un diagrama Gantt para visualizar los tiempos y las tareas planificadas.

Capítulo 9: Conclusiones y futuras mejoras. En el último capítulo se dan a conocer los problemas encontrados, las conclusiones y las futuras mejoras.

Capítulo 2: Marco regulador

En todo proyecto es importante conocer el marco regulador del mismo. Es decir, es importante conocer que leyes generales y normas pueden influenciar en el desarrollo y ejecución del proyecto.

2.1. Análisis de la legislación aplicable sobre la implementación

La legislación es un asunto que debe ser estudiado para ver en qué puede ayudar o frenar el desarrollo del proyecto. En esta sección estará estructurada en:

- Riesgos (del proyecto, técnicos y del negocio).
- Responsabilidades profesionales.
- Responsabilidades éticas.
- Riesgos laborales.
- Privacidad y seguridad.
- Otros: Soporte jurídico, Compensación económica o Supervisión.

2.1.1. Riesgos

Cuando se habla de riesgo, se entiende por la probabilidad de que suceda un evento, impacto o consecuencia adversos.

Para tener una visión completa es necesario listar los posibles factores de riesgo. Es decir, cuáles circunstancias o situaciones que aumenta las probabilidades de que suceda un evento. Se diferencia en las siguientes:

- Riesgos del proyecto.
- Riesgos técnicos.
- Riesgos del negocio.

2.1.1.1. Riesgos del proyecto

Los riesgos asociados a la realización del proyecto son los más importantes. Se enumeran a continuación:

- Los riesgos asociados al proyecto.
- Acciones para mitigar o eliminar ese riesgo.
- Criticidad del riesgo.

Tipo de riesgo	Riesgo	Acciones	Criticidad
Presupuesto	Falta de presupuesto	Tener un fondo de contingencia para el posible caso de que faltara presupuesto.	Baja
Planificación	Falta de planificación	Conocer desde el primer momento las distintas fechas para poder planificar. Añadir un par de días para cada fase por si hubiera un problema en cualquier momento.	Media
Recursos	Necesidad de recursos eficientes o de más recursos	Desarrollar una aplicación requiere unos recursos bastante limitados (un ordenador con una especificación media podría valer) y acceso a internet.	Baja
Personal	Más personal o personal mejor cualificado	Mientras más personal exista o mejor cualificado sea, eliminará riesgos. El objetivo del proyecto es centrarlo en un trabajo que pueda ser desarrollado por una persona.	Baja
Requisitos	Falta de requisitos y poca especificación de los requisitos	Establecer reunión semanales/mensuales. Realizar demostraciones del proyecto para que el cliente pueda dar feedback.	Baja

Tabla 1 - Tabla de Riesgos del Proyecto

2.1.1.2. Riesgos técnicos

Los riesgos técnicos son aquellos que ponen en peligro la calidad del proyecto. En caso de que exista algún riesgo técnico conllevará a que el proyecto sea más complejo de lo estimado. En este caso, se evalúan los riesgos que están relacionados:

Tipo de riesgo	Riesgo	Acciones
Requisitos	Falta de definición de requisitos.	Aumentar reuniones con cliente si existe una falta de definición de requisitos.
Diseño	Arquitectura no adaptable. Escalabilidad de la solución.	Estudiar distintas soluciones a aplicar y buscar una solución que sea escalable.
Implementación	Falta de personal. Falta de tiempo.	Planificar el tiempo dedicado a las diferentes tareas.
Interfaz	Interfaz poco usable.	Mejorar la interfaz con las interacciones del usuario.
Verificación	Falta de pruebas en el sistema.	Definir un listado de pruebas que validen la aplicación y la ejecución de las pruebas.
Incertidumbre técnica	Desconocimiento de las herramientas técnicas usadas.	Realizar cursos o tutoriales si hay alguna herramienta o tecnología que no se domine.

Tabla 2 - Tabla de Riesgos técnicos

Se puede observar que muchos riesgos técnicos son también riesgos del proyecto. Es importante destacar que están estrictamente relacionados ya que es la gestión e implementación técnica de un proyecto, por lo que muchos riesgos son iguales o similares.

2.1.1.3. Riesgos del negocio

Estos riesgos, a diferencia de los anteriores, implican que si se producen alguno de ellos es posible que el proyecto se cancele.

Los riesgos que se han observado de negocio son los siguientes:

- Utilidad de servicio. El uso que se le pueda dar a la aplicación. Si nadie necesita la aplicación no tiene sentido hacer el proyecto. El proyecto será usado por estudiantes de la asignatura de "Dirección de Proyectos Software".

- Dificultad de venta. El producto o servicio sea difícil de vender. En este caso, el producto no está enfocado en ser vendido pero sería fácil crear licencias de uso de la aplicación.

- Apoyo. En este caso, si el proyecto carece de apoyo, ya sea por el cliente o por el equipo no tendrá sentido hacerlo.

- Presupuesto. La falta de presupuesto podría generar problemas en el proyecto. Por eso, el proyecto ha sido realizado teniendo en cuenta los diferentes gastos que podrían existir.

2.1.2. Responsabilidades profesionales

Las responsabilidades profesionales que se derivan de la realización de este proyecto serían las siguientes:

- Aplicación segura. La aplicación debe ser segura siendo recomendable al existir una comunicación entre servidor y usuario debiendo utilizar el protocolo HTTPS (Hypertext Transfer Protocol Secure).

- Detección de problemas. Poder monitorizar de forma correcta para que en el caso de que se produzcan eventos de fallos minizar los fallos que se produzcan en el servidor.

- Actualizaciones. En el caso de que la aplicación fuera vendida en forma de licencia, se podría dar de forma gratuita las actualizaciones menores y los arreglos para fallos de seguridad.

- Copia de seguridad. En el caso de que la base de datos se corrompiera o sufriera un ataque por un agente externo, sería interesante realizar copias de seguridad de la aplicación.

- Encriptación de la base de datos. La base de datos si contuviera datos personales sería conveniente que estuviera encriptada.

- Cumplir con la Ley Orgánica de Protección de Datos (LOPD).

2.1.3. Responsabilidades éticas

En este caso, las responsabilidades éticas sería no hacer un uso fraudulento de los datos que se recopilen. El sistema de gamificación para la gestión de proyectos software no tendrá documentos que tengan información de empresas o similar. Los datos más críticos serán los del usuario. La contraseña suele ser el dato más crítico, pero en este caso, al generarse de forma aleatoria no supone un problema.

2.1.4. Riesgos laborales

Para esta aplicación no existe un riesgo laboral directo. Sólo se han detectado una serie de riesgos asociados al proyecto, técnicos y del negocio en el apartado 2.1.1. Riesgos.

2.1.5. Privacidad y seguridad

La privacidad y seguridad es un apartado importante para no comprometer la confidencialidad del usuario. Para proteger se deberían tomar las siguientes consideraciones:

- La base de datos deberá estar encriptada. Los datos más críticos como la contraseña con funciones hash OWHF (One-Way Hash function). Es decir, que solo se tengan un solo sentido.
- Implementar el protocolo HTTPS usando un certificado (que sea emitido por una entidad de certificación raíz o intermedia de confianza) en el servidor.
- Cumplir con la Ley Orgánica de Protección de Datos (LOPD).

2.1.6. Otros: Soporte jurídico, compensación económica o supervisión

Esta aplicación al ser un sistema de gamificación para estudiantes y no buscar un fin económico (ya que en un principio no sería vendido a terceras partes), ya que el uso sería interno no tendría soporte jurídico alguno. Tampoco tendría compensación económica ni supervisión.

2.2. Estándares técnicos

La aplicación se ha realizado teniendo en cuenta los siguientes estándares técnicos.

2.2.1. Tecnología desarrollada e implantada

La tecnología que se ha desarrollado está basada principalmente en Laravel. Por lo que implementa un patrón de desarrollo conocido como MVC:

- Modelo: capa de acceso a los datos de una base de datos.
- Vista: capa de presentación de los datos procesados por un controlador.
- Controlador: capa de negocio donde se crean modelos para realizar peticiones a la base de datos, se procesan los datos y se presentan en una vista.

Laravel además tiene:

- Inyección de métodos.
- Sistema de rutas.
- Cacheado de rutas.
- Autenticación.
- Múltiples bases de datos usando un mismo modelo.
- Múltiples sistemas de ficheros (S3, local, etc).

2.2.2. Lenguajes de programación

Los lenguajes de programación que se han utilizado han sido los siguientes para las distintas capas de la aplicación:

- Lógica de negocio: PHP (versión 7.0) siendo el framework de desarrollo Laravel (haciendo uso del patrón MVC).
- Interfaz de usuario: HTML5, Javascript y hojas de estilos CSS.

2.2.3. Herramientas utilizadas

Las herramientas que se han utilizado para desarrollar han sido las siguientes:

- IDE desarrollo: Sublime Text.

- Gestión de dependencias:
    - Composer para PHP.
    - Bower para Javascript y CSS.
- Repositorio de código fuente: GIT (servidor en la nube Bitbucket).
- Navegadores usados para visualización de la interfaz de usuario: Google Chrome, Firefox e Internet Explorer.
- Base de datos: MySQL.
- Gestor de base de datos: MySQL Workbench 6.3 CE.
- Sistemas operativos utilizados: Windows 10 y Ubuntu.
- Servidor web: Apache Tomcat.

2.3. Propiedad intelectual de la idea

En este apartado se discutirá sobre la propiedad intelectual de la idea.

2.3.1. Patentabilidad

En este caso, la patentabilidad de la idea al ser un proyecto sin ánimo de lucro no tendría sentido realizar una patente del mismo. En mi opinión, crear una patente lo único que podría hacer disminuir el uso de la aplicación. Ya que se añadiría un coste al desarrollo de la aplicación software.

Quizás sería interesante en realizar un enfoque “open source”. Es decir, hacer que el código sea libre y abierto (considerar el proyecto como FOSS: “Free and Open Source Software”). Las implicaciones que podría tener serían las siguientes:

- Seguridad: Mientras más desarrolladores puedan ver el código, más número de defectos pueden encontrar (incluso pueden desarrollar el parche de ese defecto).
- Calidad: Mientras más personas usan el software, más se acerca al uso de los potenciales usuarios. Esto conlleva a que el acercamiento de las necesidades sea mucho más cercano que una aplicación con el código cerrado.

- Además de las anteriores citadas, también se podrían añadir: flexibilidad, interoperabilidad, bajo o nulo coste, etc.

2.3.2. Protección

Como se ha comentado en el apartado anterior, sería más interesante que, en vez de proteger el código, liberarlo para poder ofrecer mayor seguridad, calidad, bajo coste, etc. al usuario final.

En todo caso, la lógica de negocio está en el lado del servidor por lo que en caso de plagio, tendría que implementarse una lógica de negocio similar (o en la medida de lo posible igual).

Capítulo 3: Entorno socio-económico

Hay que destacar que en la elaboración de un proyecto, en este caso, siendo un Trabajo de Fin de Grado, es de clara utilidad conocer el entorno socio-económico que lo rodea. Es decir, conocer qué impacto social, económico, medioambiental, ético, etc. que podría aportar el proyecto.

Además debe realizarse un presupuesto de la elaboración del Trabajo de Fin de Grado para que con exactitud pueda valorarse de forma económica.

3.1. Presupuesto de la elaboración del trabajo

El presupuesto de la elaboración del Trabajo de Fin de Grado se va a dividir en distintos tipos de gastos. Al final de la sección se realizará un resumen completo de forma que se visualice los distintos totales.

3.1.1. Coste personal

Se ha definido distintos roles que podrían ser los encargados para realizar este Trabajo de Fin de Grado. Siendo los siguientes:

- Jefe de proyecto: Encargado de liderar el proyecto, tomar decisiones técnicas estratégicas y llevar a buen puerto el proyecto.
- Analista funcional: Es el rol que hace de vínculo entre el usuario y la parte informática. Controla, analiza y supervisa el desarrollo funcional de las aplicaciones.
- Analista programador: Persona que se encarga de tener funciones de analista técnico y de un programador. Es decir, partiendo de la información que le provee el analista funcional se encarga de desarrollar las aplicaciones.
- Ingeniero de QA (Quality Assurance): Especialista en la realización de planes de prueba y de ejecución de los mismos para asegurar que el software cumple determinada calidad.

Para calcular el coste aproximado de cada miembro se ha hecho uso de distintos portales de ofertas de empleo (LinkedIn, Infojobs, etc) y se ha realizado una media aproximada.

Para estimar cuántas horas ha dedicado cada rol está basado en la dedicación en el Trabajo de Fin de Grado y en la planificación (diagrama de Gantt).

Rol	Empleados	Coste por hora (€/h)	Horas	Coste (€)
Jefe de proyecto	1	35,00 €/h	30	1.050,00 €
Analista funcional	1	25,00 €/h	120	3.000,00 €
Analista programador	1	22,00 €/h	230	5.060,00 €
Ingeniero QA	1	25,00 €/h	60	1.500,00 €
		Total	**440**	**10.610,00 €**

Tabla 3 – Tabla Coste total personal

El coste total de personal es de 10.610,00 €.

3.1.2. Coste recursos o material

Además de conocer el coste de personal, es necesario conocer el coste de los recursos o materiales para poder llevar a cabo el Trabajo de Fin de Grado.

Se va a dividir en dos secciones dependiendo de la etapa en la que se haga uso:

- Etapa de desarrollo: Etapa en la que se va a realizar el desarrollo de la aplicación y no va a requerir que el usuario pueda hacer uso del mismo.
- (Posible) Etapa de puesta en producción: Etapa en la que la aplicación provee un servicio al usuario y es de debido cumplimiento que pueda dar uso a la aplicación.

Los recursos que se han utilizado en este Trabajo de Fin de Grado no son exclusivos para el mismo, si no, que pueden servir para otros proyectos, por lo que no se puede imputar el coste completo al Trabajo de Fin de Grado.

En este caso, se aplicará una amortización de los activos comprados. El método de amortización que se va a realizar es el de amortización constante o lineal. Es decir, cada año se aplica la misma cuota de amortización.

Por simpleza, el tiempo de uso que se les ha dado a los recursos ha sido de un año (se puede comprobar en 8.2. Diagrama de Gantt).

En la etapa de desarrollo se han tenido en cuenta los siguientes costes:

Recurso	Cantidad	Coste (€)	Vida útil	Coste asumido (€)
Equipo de desarrollo (Gama media)	1	700 €	3 años	233,34 €
Sublime Text	1	67,35 € (80 $ al cambio actual)	2 años*	33,68 €
Apache Tomcat	1	0 €	-	0 €
MySQL	1	0 €	-	0 €
			Total	**267,12 €**

Tabla 4 - Tabla de Coste Material (Etapa desarrollo)

* Es destacable que la licencia de "Sublime Text" es una licencia permanente por usuario. Es decir, que tendría una alta vida útil. El problema que exista es que las nuevas versiones del software no se incluyen en la licencia y habría que actualizar por un precio reducido.

En la etapa de puesta en producción se han tenido los siguientes costes:

Recurso	Cantidad	Coste (€)	Vida útil	Coste asumido (€)
Servidor (Gama media)	1	1400 €	4 años	350,00 €
MySQL	1	0 €	-	0 €
Apache Tomcat	1	0 €	-	0 €
			Total	**350,00 €**

Tabla 5 - Tabla de Coste Material (Etapa producción)

Para el presupuesto se ha tenido en cuenta la posible puesta en producción por lo que se ha añadido al coste total del presupuesto.

El coste total de recursos es de un total de 617,12 €.

3.1.3. Resumen presupuesto

Teniendo en cuenta que se pone en producción resumen del presupuesto se presenta en la siguiente tabla:

Tipo de gasto	Coste (€)
Personal	10.610,00 €
Recursos y materiales (desarrollo)	267,12 €
Recursos y materiales (producción)	350,00 €
Total	**11.227,12 €**

Tabla 6 - Tabla Resumen Presupuesto

Para calcular el presupuesto habría que añadir los siguientes factores al presupuesto:

- Margen de riesgo.
- Margen de beneficio.
- Impuesto al valor añadido (IVA).

En este caso, la aplicación se ha desarrollado sin ánimo de lucro, por lo que no se ha tenido en cuenta estos factores. El coste total del presupuesto es de 11.227,12 €.

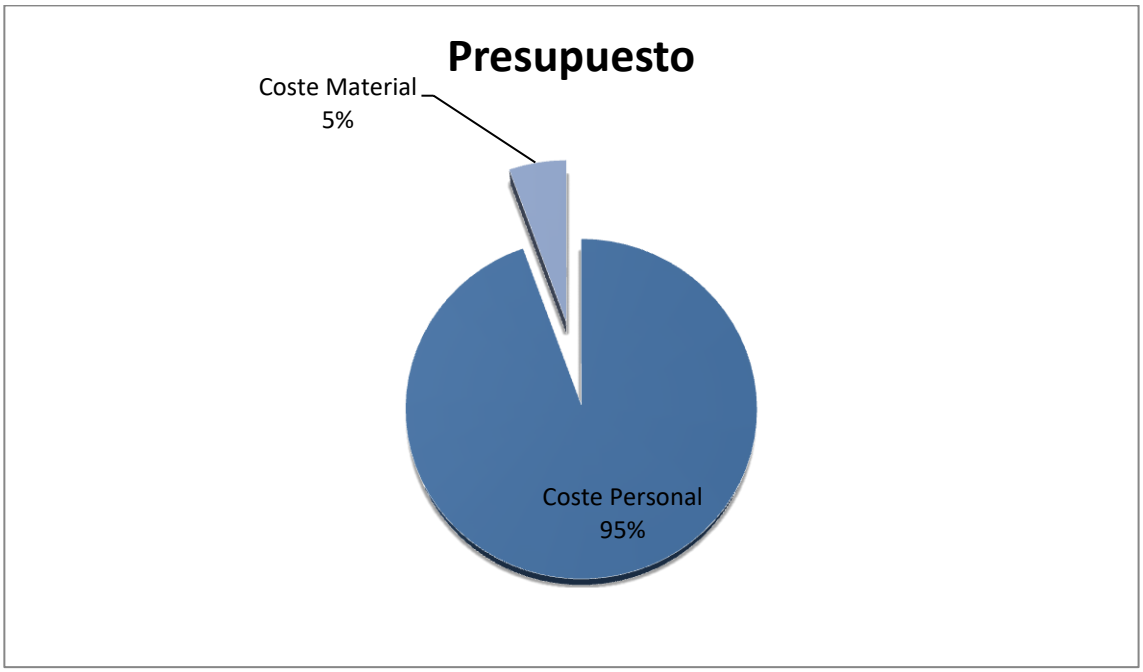

Ilustración 1 - Visualización del Presupuesto

En la gráfica anterior podemos ver que el coste del personal es mucho mayor (95%) que el coste del material (5%).

3.2. Impacto socio-económico

Conocer el impacto socio-económico es importante ya que es un indicador que permite medir el progreso y comparar la eficacia de un Trabajo de Fin de Grado. Es importante realizar un análisis y saber qué impacto podría ser el esperado (aunque este indicador fuera negativo).

3.2.1. Impacto económico

El Trabajo de Fin de Grado el impacto económico esperado se ha evaluado de la siguiente forma:

- Si es ofrecido como un servicio extra a los alumnos para la realización de una asignatura. En este caso, tendrá un impacto indirecto ya que los alumnos podrán aprender más de la asignatura y verán más valorado sus estudios universitarios. Lo que podría llevar a que la universidad tuviera más reputación y más excelencia en los estudios.

- Se podría ofrecer en forma de licencia de uso como una plataforma para otras universidades o academias de negocios (para la realización de alguna posible MBA enfocada a desarrollo software), etc. En este caso, tendría un impacto directo ya que sería el coste de la licencia.

La aplicación se podría ofrecer de dos formas distintas:

- On premise: Es decir, el cliente lo tendría instalado en su propio servidor.

- Cloud: La aplicación estaría en la nube y el cliente haría un uso de la aplicación.

El coste de la licencia podría verse valorado por los siguientes aspectos:

- Número de usuarios en la aplicación (incremental por cada mayor número de usuarios).

- Uso de la aplicación: Mientras más funcionalidades tenga, mayor coste.

- Dependiendo de si es “on premise” o “cloud”. Normalmente al ser “on premise” es una licencia anual independiente del uso que se le vaya a dar (ya que es el cliente quien tendrá que adecuar los requisitos a los usuarios que tenga) y en cloud por uso de la aplicación (número de usuarios o funcionalidades disponibles). Incluso sería valorable el realizar pago de la licencia de forma mensual (y si el cliente realizase pago de forma anual, realizar un descuento en la licencia). Con ello incentivaríamos una duración de licencias más largo y siendo el proyecto más sostenible.

También implementar un modelo “Freemium” podría ser un posible aspecto interesante a desarrollar. Este modelo se comentará en el apartado 3.2.5. Plan de explotación.

3.2.2. Impacto social

El impacto social cada vez toma más importancia en la realización de proyectos, ya que no sólo es conseguir beneficios (ser rentable y alcanzar buenos resultados), si no poder ayudar y hacer un cambio real a la sociedad.

En este caso, el impacto social que se podría esperar del Trabajo de Fin de Grado propuesto sería más bien bajo. Es decir, los efectos que este Trabajo de Fin de Grado plantea sobre la comunidad en general serían difíciles de observar.

Lo único destacable que se podría observar es que los estudiantes que después de estar usando la aplicación hayan aprendido los siguientes valores:

- Constancia en la gestión de proyectos. El uso día a día permite llevar el control del proyecto, conocer qué riesgos se están asumiendo y qué es lo que se está planificado.

- Sinceridad con uno mismo y con el proyecto. Si el proyecto sale mal, sea cual sea el motivo, el estudiante aprenderá de sus errores y querrá mejorar en base a sus conocimientos y a su experiencia. Es decir, será una persona en el futuro con mayor madurez y preparado para el error (y para triunfar, dicho sea de paso).

- Mejora en el trabajo en equipo. Hay ciertos objetivos que requieren de distintos profesionales para llegar a buen puerto. En este caso, la gestión de proyectos software es una buena propuesta para mejorar en el trabajo en equipo. Es decir, mejorar la comunicación entre los distintos miembros, división de tareas entre los distintos miembros, etc.

3.2.3. Impacto medioambiental

Esta aplicación no tiene un impacto directo en el área medioambiental. Solo se han tenido en cuenta las siguientes consideraciones:

- Coste de luz del servidor: La aplicación tiene requisitos mínimos para un uso intensivo de alrededor 50 alumnos y 5 profesores. Por lo que sería suficiente con un servidor de gama media-baja.

- Subida de documentos de forma electrónica: La aplicación permite que los estudiantes suban sus documentos de forma electrónica reduciendo la impresión a papel de documentos.

3.2.4. Impacto ético

El Trabajo de Fin de Grado no se ha centrado en conseguir un impacto ético en la sociedad. Por lo que el impacto ético esperado es de cero. Las únicas puntualizaciones que se han tomado y quizás podrían ayudar a promover unas actitudes más correctas del comportamiento humano serían los posibles valores que podría obtener un estudiante (en el apartado 3.2.2. Impacto social).

3.2.5. Plan de explotación

Al realizar un plan de explotación es importante saber que toda la información son previsiones. Es decir, se trata de demostrar que:

- Disponemos de gran conocimiento del mercado al que se dirige el producto o servicio.
- Sabemos la forma de llegar en el público objetivo y convertirlos en clientes.

Es importante destacar que el sector de la gamificación está teniendo una progresión cada vez mejor y un gran crecimiento anual (sobre todo, en las campañas de gamificación del cliente).

En este caso se podría realizar con distintas propuestas:

- Propuesta Freemium. Una aplicación de tipo “Freemium” consiste en que parte de la aplicación sería de uso gratuito pero, a cambio, tendría publicidad visual. Además de una parte “Premium” en la que pagando una cierta cantidad de dinero de forma mensual (suscripción mensual) la publicidad sería eliminada y podría tener acceso a más funcionalidades.
- Propuesta licencia. Como se ha comentado en el apartado 3.2.1. Impacto económico, esta licencia podría estar enfocada a cómo se instala la aplicación y el uso que se haría de la mismo.

3.2.6. Consideraciones sobre aspectos económicos

Es importante destacar que en la realización de este Trabajo de Fin de Grado se ha enfocado no en el aspecto económico, sino en la creación de un sistema de gamificación para que los estudiantes puedan maximizar su aprendizaje y la aplicación de

conceptos teóricos a la realidad. Es decir, la máxima ha sido la mejora del aprendizaje y el refuerzo de conocimientos sobre gestión de software.

Capítulo 4: Contexto y Definición del Problema

En este apartado se va a describir de forma básica la Dirección de Proyectos de desarrollo Software y qué procedimientos se van a llevar para realizar de forma correcta la gestión y dirección del proyecto. Se va a enfocar en la asignatura del Grado en Ingeniería Informática llamada: “Dirección de Proyectos de desarrollo Software”.

4.1. Dirección de Proyectos de desarrollo Software

4.1.1. Oferta de Prestación de Servicios

La Oferta de Prestación de Servicios tiene como objetivo demostrar que se ha entendido lo que el cliente quiere/desea. Se ha identificado el problema del cliente y se propone una solución.

En este documento se va a hablar sobre:

- Objetivo de la propuesta.
- Finalidad del trabajo a realizar.
- Objetivo del sistema a desarrollar.
- Método de trabajo
- Descripción de la metodología a utilizar.
- Equipo de trabajo (estimación de recursos, labores de cada persona y organigrama, incluido el del cliente).
- Organización del trabajo.
- Planificación (Gantt).
- Recursos.
- Tecnología.
- Presupuesto.

Para calcular los costes será necesario tener un documento interno de cálculo de costes (DCC). Después de calcular los costes habrá que aplicar un margen de beneficio de los costes, primas de riesgo, etc.

La fórmula del presupuesto total sería la siguiente: (Coste de las diferentes partidas + Beneficio + Riesgo + IVA).

4.1.2. Gestión de Configuración

La gestión de la configuración del software (conocido como SCM) es un proceso que soporta el ciclo de vida del software. Este proceso ayuda a mejorar la gestión de proyectos, actividades de desarrollo y mantenimiento, etc.

Los propósitos del documento son los siguientes:

- Aumentar la calidad del software.
- Aumentar la productividad: crear más y mejor en menos tiempo.
- Reducir los errores que se produzcan.
- Control y registro de los cambios del sistema.
- Evitar problemas que se deriven de sincronizar de forma incorrecta los cambios.

Se producen cambios en los requisitos, en el equipo de trabajo, en el cliente, en los plazos, etc. Pero eso no hace que sea importante mantener la integridad en todos los productos del proyecto.

4.1.3. Gestión de Calidad

El documento de Gestión de Calidad es importante ya que tiene como propósito la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos. Además de existir una concordancia con los estándares de desarrollo explícitamente documentos junto con las características implícitas que se espera de todo software desarrollado profesionalmente [1].

Los procesos de gestión de calidad de software deben enfocarse a:

- Cómo los productos software van a satisfacer al cliente.
- Cómo proporcionar valor a los clientes y demás personal implicado.

La conclusión sería proveer la calidad de software precisa para conocer los requisitos del software.

Las tareas comunes que se van a llevar en todas las fases:

- Gestión de expectativas del cliente.
- Gestión de los riesgos.
- Gestión de relaciones con el cliente.
- Gestión interna del personal.

4.1.4. Gestión de Proyectos

La finalidad de la gestión de proyectos es planificación, seguimiento y control de las actividades, los recursos humanos y materiales que intervienen en el desarrollo de un Sistema de Información.

Las actividades que se realizan son las siguientes:

- Inicio del proyecto: Actividades de inicio del proyecto.
- Desarrollo del proyecto: Actividades de seguimiento y control.
- Cierre del proyecto: Actividades de finalización del proyecto.

4.1.5. Estudio de Viabilidad del Sistema

El objetivo de este documento es realizar un análisis de un conjunto concreto de necesidades para proponer una solución a corto plazo que tenga en cuenta restricciones económicas, técnicas, legales y operativas.

Se realizan las siguientes tareas:

- Los requisitos que se tienen que satisfacer.
- Estudio de la situación actual.
- Identificación de alternativas de la solución.
- Valoración de las soluciones y selección de la que más valor aporte.

Existen dos tipos de requisitos:

- Requisitos de Usuario (EVS)

- Requisitos de capacidad.
- Requisitos de restricción.

- Requisitos del Sistema (ASI):
  - Requisitos funcionales.
  - Requisitos de rendimiento.
  - Requisitos de Interfaz.
  - Requisitos de Operación.
  - Requisitos de recursos.
  - Requisitos de verificación.
  - Requisitos de pruebas de aceptación.
  - Requisitos de documentación.
  - Requisitos de seguridad.
  - Requisitos de calidad.
  - Requisitos de daño.
  - Requisitos de mantenibilidad.
  - Requisitos de recuperación entre fallos.

4.1.6. Análisis de los sistemas de información

En este caso el objetivo es poder obtener de una especificación detallada del sistema de información que satisfaga las necesidades información de los usuarios y sirva de base para el posterior diseño del sistema.

Los puntos más importantes son:

- Definición del sistema:
  - Determinar el alcance del sistema.
  - Identificar el entorno tecnológico.
  - Especificación de estándares y normas.
  - Identificación de usuarios participantes.

- Establecimiento de requisitos:
    - Obtener los requisitos.
    - Especificar casos de uso.
    - Análisis de requisitos.
    - Validación de requisitos.

4.1.7. Diseño de los sistemas de información

En este documento se define la arquitectura del sistema y del entorno tecnológico que le va a dar soporte, junto con la especificación detallada de los componentes del sistema de información.

4.1.8. Pruebas

Las pruebas son necesarias para validar en distintos momentos del ciclo de vida de software para comprobar la calidad del software:

- Funcionamiento correcto de un módulo.
- Funcionamiento correcto entre distintos módulos.
- Funcionamiento completo operativo (prueba completa del sistema).
- Aplicar cambios nuevos no modifiquen cambios sobre un módulo que funcionaba bien antes.

El objetivo de las pruebas es:

- Verificar que estamos construyendo correctamente el producto.
- Validar que estamos construyendo el producto correcto.

Hay diferentes tipos de pruebas y son las siguientes:

- Pruebas unitarias.
- Pruebas de integración.
- Pruebas del sistema.
- Pruebas de implantación.

- Pruebas de aceptación.
- Pruebas de Regresión.

4.1.9. Implantación y Aceptación del Sistema

El objetivo es la entrega y aceptación del sistema en su totalidad. Además de realizar las actividades necesarias para el paso a producción del sistema.

Las fases que tiene este apartado son las siguientes:

- Establecimiento del Plan de Implantación.
- Formación Necesaria para la Implantación.
- Incorporación del sistema al entorno de operación.
- Carga de datos al entorno de operación.
- Pruebas de implantación del sistema.
- Pruebas de aceptación del sistema.
- Preparación mantenimiento del sistema.
- Establecimiento del acuerdo de nivel de servicio.
- Presentación y aprobación del sistema.
- Paso a producción.

4.1.10. Documento Histórico de Proyecto

El objetivo de este documento es recoger toda la información relevante que se ha ido recopilando en diversos documentos a lo largo del ciclo de vida del proyecto. Además facilitar la consulta para aprovechar la experiencia adquirida en proyectos posteriores.

4.2. Problema planteado

El problema planteado es realizar un sistema de gamificación basado en la asignatura “Dirección de Proyectos de desarrollo Software”. Es decir, es crear una

gamificación para que los estudiantes de la asignatura puedan aprender de una forma más asociada a los juegos. En el apartado 4.3. Gamificación en el ámbito educativo se puede ver qué ventajas aporta como técnica de aprendizaje.

En la asignatura "Dirección de Proyectos de desarrollo Software" el alumno aprende a cómo gestionar y dirigir un proyecto software. Este caso se compone de poder crear un equipo donde los alumnos pueden trabajar entre ellos para poder realizar un proyecto de desarrollo Software cumpliendo los dos objetivos básicos:

- Terminar el proyecto sin exceder el presupuesto.
- Terminar el proyecto en el tiempo planeado.

Estos dos objetivos se verán afectados por eventos que puedan ocurrir, adquisición de material para realizar el proyecto (tener una oficina, sillas, mesas, ordenadores, etc.), envío de documentos al cliente, etc.

4.3. Gamificación en el ámbito educativo

La gamificación es una técnica de aprendizaje que traslada la mecánica de los juegos al ámbito educativo y al ámbito profesional. Con esto, se consigue mejores resultados para mejorar conocimientos y aprender nuevas habilidades.

Cada vez gana más terreno en las metodologías de formación debido a:

- Facilita la interiorización de conocimientos de una manera más amena.
- Genera una experiencia positiva en el usuario.
- Motiva a los alumnos y consigue mayor compromiso de las personas.
- Incentiva el ánimo de superación.
- Recompensa al usuario en función de los objetivos alcanzados.

Las técnicas más utilizadas son las siguientes:

- Obtener premios a medida que se consiguen diferentes objetivos.
- Uso de clasificación para destacar a los mejores jugadores (o equipos) en un ranking.
- Escalado de niveles. Se definen una serie de niveles y el usuario debe ir superando para llegar al siguiente nivel.

- Acumulación de puntos. Los puntos se consiguen por realizar determinadas acciones y se van acumulando a medida que se van realizando más y más acciones.

Capítulo 5: Análisis del sistema

El análisis del sistema es el capítulo donde se evalúan las necesidades del cliente. El objetivo final es conocer cuáles son los requisitos del sistema y el cliente se encuentre satisfecho.

5.1. Definición de los requisitos del sistema

A continuación se va a definir los requisitos del sistema. Los requisitos del sistema serán definidos haciendo uso de Behaviour-driven Development [3] (BDD). Esta forma se ha impuesto en muchas empresas software en las que se ha impuesto el desarrollo ágil.

5.1.1. Behaviour-driven Development (BDD)

Behaviour-driven Development [3] (BDD) o desarrollo guiado por el comportamiento es un proceso de desarrollo de software que ha evolucionado del Test-driven Development (TDD). BDD es un mecanismo que haciendo uso de una plantilla y respondiendo a una serie de aspectos técnicos, ayuda a los tres enfoques (negocio, desarrollo y calidad).

Behaviour-driven Development es una idea de cómo se pueden unir:

- Intereses de negocio: respondiendo a las siguientes preguntas: ¿Qué funcionalidad necesita el usuario? ¿Qué beneficio tiene?

- Aspectos técnicos: en este caso, serán estas preguntas que tendrá que responder: ¿Qué contexto inicial debe existir? ¿Qué debe ocurrir? ¿Qué salida se espera?

En Behaviour-driven Development se utiliza un domain-specific language o lenguaje simple específico de dominio (también conocido como DSL). Esto conlleva a que se construyan frases en un formato sencillo y fácil de entender.

5.1.2. Antecedentes al BDD

Para conocer los antecedentes, habría que volver partir de la definición de los caso de uso (en inglés, conocidos como use case). En la toma de requisitos de software, un caso de uso es una descripción generalizada de un conjunto de interacciones entre el sistema y uno o más actores, donde el actor puede ser un usuario u otro sistema.

En metodologías de desarrollo ágiles, se definen las historias de usuario como una descripción de una o más frases en un lenguaje de negocio una necesidad de un usuario del sistema como parte de la función de su trabajo. Lo importante de las historias de usuario es que se obtienen de una forma concisa:

- **Quién:** El usuario que interactúa con el sistema.
- **Qué:** Qué acción va a tomar ese usuario.
- **Por qué:** Definición del motivo o motivos de ese usuario.

A veces, las historias de usuario tienen un detalle limitado de las necesidades. Las historias de usuario no son inmutables, pero esto no significa que se tengan que cambiar en cualquier instante. Es recomendable dedicar un tiempo a cada historia de usuario para que no quede nada pendiente ni detalles de libre interpretación. Todos los detalles que se puedan interpretar de distintas formas pueden llevar a:

- Un mal desarrollo por parte del desarrollador.
- Una funcionalidad que no aporta valor al negocio.
- Una funcionalidad en la que no se pueden ejecutar pruebas para validar que funciona correctamente.

No todas las historias de usuario son igual de importante y no todas deberían ir a un mismo saco.

Una recomendación al trabajar con historias de usuario es asignarlas un nivel de prioridad. Para un proyecto básico, podría existir la siguiente clasificación:

- Prioridad alta: Una historia de usuario con prioridad alta tiene un porcentaje muy alto de que estará implementada en el Producto Mínimo Viable (MVP).
- Prioridad media: Este tipo de historias de usuario serán hechas después de las de prioridad alta y seguramente complementarán el funcionamiento de las historias de usuario ya realizadas. Un ejemplo fácil de entender. Para poder gestionar usuarios (realizar las operaciones básicas: crear, listar, buscar, editar y borrar) es necesario primero que la entidad usuario exista.

- Prioridad baja: Una historia de usuario que podría no figurar en el Producto Mínimo Viable (MVP).

Es importante saber que en muchas ocasiones hay una historia de usuario que está relacionado con otra. Es recomendable añadir la relación que tienen entre ellas:

- Puede estar simplemente relacionada a otra (muchas veces aporta información complementaria o que aporta un conocimiento más completo de negocio al revisar la historia de usuario relacionada).

- Puede bloquear otra historia de usuario. Esto significa que una historia de usuario debe realizarse antes de empezar con la historia de usuario que está bloqueando.

- Puede ser duplicada de otra historia de usuario. En este caso, se suele mezclar las dos historias de usuario y actualizar las relaciones que tuviera.

Un listado de ejemplos de descripciones de historias de usuario sería:

"Como administrador, para poder gestionar los usuarios del sistema necesito poder listar los usuarios."

"Como analista de negocio, para poder gestionar las facturas de la empresa necesito poder listar las facturas."

"Como auditor, para poder gestionar la contabilidad de una empresa necesito poder listar los gastos producidos de forma mensual."

5.1.3. Convenciones

Para poder trabajar de una forma más uniforma es recomendable usar (y si no existe definir) una serie de convenciones para poder trabajar de una forma correcta con Behaviour-driven Development:

- Normalmente, el lenguaje suele ser el inglés ya que suele ser el idioma común y es utilizado en entornos internacionales y donde los miembros son de distintas nacionalidades. Para este Trabajo de Fin de Grado, por simpleza, se utilizará el idioma español.

- Las funcionalidades es recomendable que estén dentro del control de versiones para poder tener un versionado de las funcionalidades y ver qué se ha modificado con cada cambio. La extensión recomendable es ".feature".

5.1.4. Estructura

La estructura de una funcionalidad es bastante sencilla y consta de los siguientes elementos:

- Título de la funcionalidad: Representa de forma rápida de qué trata la funcionalidad. El nombre del archivo suele ser el mismo pero en minúsculas y con guiones. Ejemplo: “Listar usuarios” y el nombre de fichero: “listar-usuarios.feature”.

- Breve descripción de la funcionalidad: Es una forma de añadir una documentación con valor para negocio y desarrollo. Además de que ayuda a conocer quién usará la funcionalidad y que valor obtendrá. Son tres elementos

    - Beneficio ¿Qué valor de negocio obtengo?

    - Usuario: ¿Qué rol de negocio o de proyecto disfrutará del {beneficio}?

    - Funcionalidad: ¿Qué necesidad tiene el {usuario}?

- Un listado de escenarios. Un escenario es un lugar donde se desarrolla una acción o suceso. Una forma rápida de entender qué son es asimilarlos a los criterios de aceptación. Dicho de otra forma, son una descripción de cada caso específico de la funcionalidad. Todos los escenarios tienen la siguiente estructura:

    - Dado {contexto inicial}: Se define el contexto inicial en el que se producirá un evento.

    - Cuando {evento}: El momento en el que ocurre el evento.

    - Entonces {salida esperada}: La salida que se espera con el evento que acaba de ocurrir.

Cabe destacar que se definirán tantos escenarios como hagan falta. El mínimo siempre será uno y el máximo cómo sea posible (y aporten un valor diferencial). Con la experiencia se mejora en la construcción de historias de usuario y la definición de los escenarios.

A continuación, he definido una plantilla que será la utilizada para la definición de las funcionalidades de este Trabajo de Fin de grado. La plantilla es la siguiente:

Título de la funcionalidad

Para {beneficio}
Como {usuario}
Necesito {funcionalidad}

Escenario: Título del escenario
Dado {contexto inicial}
Cuando {evento}
Entonces {salida esperada}

Escenario: Título del escenario 2
Dado {contexto inicial}
Cuando {evento}
Entonces {salida esperada}

Una forma sencilla de entender cómo se estructura es visualizando el siguiente ejemplo:

Enviar mensaje a otro usuario

Para poder tener un sistema donde sea posible la comunicación entre los distintos usuarios
Como usuario
Necesito poder enviar un mensaje a otro usuario

Escenario: Puedo enviar un mensaje a otro usuario
Dado que soy un usuario del sistema
Cuando envío un mensaje al usuario "Javier"
Entonces Javier recibirá el mensaje que le he enviado

Escenario: Título del escenario 2
Dado que soy un usuario del sistema
Cuando envío un mensaje al usuario "NoExiste"
Entonces recibiré un mensaje de error del sistema

5.2. Funcionalidades del sistema

5.2.1. Sesión del usuario

Iniciar sesión

Para poder tener una gestión correcta de la sesión del usuario
Como usuario
Necesito iniciar la sesión y acceder a la aplicación

Para poder iniciar sesión será necesario introducir:

- **Email**
- **Contraseña**

Un ejemplo de usuario con acceso válido será:

- **usuario@alumnos.uc3m.es**
- **password123**

Escenario: Puedo hacer login con un usuario válido
Dado que soy un usuario con acceso a la aplicación
Cuando hago login
Entonces veré la página de inicio

Escenario: No puedo hacer login con un usuario incorrecto
Dado que soy un usuario sin acceso a la aplicación
Cuando hago login
Entonces veré un mensaje de error

Escenario: No puedo hacer login si no introduzco el correo
Dado que soy un usuario sin acceso a la aplicación
Cuando hago login sin introducir un correo
Entonces veré un mensaje de error

Escenario: No puedo hacer login si no introduzco la contraseña
Dado que soy un usuario sin acceso a la aplicación
Cuando hago login sin introducir una contraseña
Entonces veré un mensaje de error

Escenario: No puedo hacer login si introduzco un email sin formato correcto
Dado que soy un usuario sin acceso a la aplicación
Cuando hago login con un email sin formato correcto
Entonces veré un mensaje de error

Cerrar sesión

Para poder tener una gestión correcta de la sesión del usuario
Como usuario
Necesito cerrar la sesión

Escenario: Puedo cerrar sesión con un usuario correcto
Dado que soy un usuario con acceso a la aplicación
Y tengo iniciada la sesión en la aplicación
Cuando cierro la sesión
Entonces veré la página de login

Duración de la sesión

Para poder tener una gestión correcta de la sesión del usuario
Como usuario
Necesito que mi sesión tenga una duración por motivos de seguridad

En este caso, la duración será de 60 minutos desde la última actividad del usuario.

Escenario: Mi sesión se cierra al pasar una duración de tiempo predeterminada
Dado que soy un usuario con acceso a la aplicación
Y tengo iniciada la sesión en la aplicación
Cuando estoy 60 minutos sin realizar ninguna acción
Entonces veré la página de login
Y mi sesión se encontrará eliminada

Una sesión por navegador

Para poder tener una gestión correcta de la sesión del usuario
Como usuario
Necesito que mi sesión sea única por sesión del navegador

Escenario: Mi sesión sólo estará abierta en un navegador
Dado que soy un usuario con acceso a la aplicación
Cuando hago login
Entonces veré la página de inicio
Y en otro navegador al dirigirme a la aplicación veré la página de login

5.2.2. Cuenta del usuario

Cambiar contraseña

Para poder tener una gestión correcta de la cuenta del usuario
Como usuario
Necesito cambiar la contraseña de mi cuenta

Para poder cambiar la contraseña será necesario introducir:

- **Contraseña actual: Para evitar que una persona con acceso a la sesión pueda cambiar la contraseña.**
- **Contraseña nueva: La contraseña a la cual quiere cambiar el usuario.**
- **Repetir contraseña nueva: Para evitar equivocaciones al escribirla.**

Escenario: Puedo cambiar de contraseña por una distinta
Dado inicio sesión con un usuario correcto
Cuando cambio de contraseña
Y vuelvo a iniciar sesión con la nueva contraseña
Entonces veré la página de inicio

5.2.3. Gestión de usuarios

Ver cuentas de los usuarios

Para poder tener una gestión correcta de los usuarios del sistema
Como administrador
Necesito ver las cuentas de los usuarios

La lista de usuario tendrá las siguientes consideraciones:

- **Se paginarán los resultados por cada 15 usuarios.**
- **Se mostrará los siguientes datos:**
  - **Nombre**
  - **Apellidos**
  - **Correo**

Escenario: Puedo listar los usuarios existentes
Dado que soy un usuario con acceso a la aplicación
Cuando entro en el listado de usuarios
Entonces veré el listado de los usuarios

Ver detalle de una cuenta de usuario

Para poder tener una gestión correcta de los usuarios del sistema
Como administrador
Necesito ver el detalle de una cuenta de usuario

Los datos que se mostrarán serán los siguientes:

- **Nombre**
- **Apellidos**
- **Correo**
- **Fecha de creación**

También permitirá realizar las siguientes acciones:

- **Editar cuenta de usuario**
- **Borrar cuenta de usuario**

Escenario: Puedo ver los detalles de un usuario
Dado que soy un usuario con acceso a la aplicación
Cuando accedo al listado de usuarios
Y veo los detalles de un usuario especifico
Entonces veré los detalles de ese usuario

Editar cuenta de usuario

Para poder tener una gestión correcta de los usuarios del sistema
Como administrador
Necesito editar una cuenta de usuario

Escenario: Puedo editar una cuenta de usuario
Dado que soy un usuario con acceso a la aplicación
Cuando accedo al listado de usuarios
Y edito una cuenta de usuario
Entonces veré el listado de usuarios con el nuevo usuario actualizado

Añadir cuenta de usuario

Para poder tener una gestión correcta de los usuarios del sistema
Como administrador
Necesito añadir una cuenta de usuario

Los datos necesarios para añadir una cuenta de usuario serán los siguientes:

- **Nombre**
- **Apellidos**
- **Correo electrónico**

Escenario: Puedo añadir una cuenta de usuario
Dado que soy un usuario con acceso a la aplicación
Cuando creo una nueva cuenta de usuario
Entonces veré en el listado de usuarios el nuevo usuario creado

Borrar cuenta de usuario

Para poder tener una gestión correcta de los usuarios del sistema
Como administrador
Necesito borrar una cuenta de usuario

No se puede borrar la cuenta del usuario que ha iniciado sesión.

Escenario: Puedo borrar una cuenta de usuario que no sea la cuenta propia
Dado que soy un usuario con acceso a la aplicación
Cuando accedo al listado de usuarios
Y borro una cuenta de usuario
Entonces dejaré de ver el usuario borrado en el listado de usuarios

5.2.4. Gestión de roles

Ver roles

Para poder tener una gestión correcta de los roles del sistema
Como administrador
Necesito ver un listado de los roles

Por defecto, habrá dos roles creados:

- **Administrador**
- **Estudiante**

Escenario: Puedo listar los roles existentes en la aplicación
Dado que soy un usuario con acceso a la aplicación
Cuando entro en el listado de roles
Entonces veré el listado de roles

Ver miembros de un rol

Para poder tener una gestión correcta de los roles del sistema
Como administrador
Necesito ver los miembros que tiene un rol

Escenario: Puedo listar los miembros que tengan un rol un asignado
Dado que soy un usuario con acceso a la aplicación
Cuando entro en el listado de roles
Y accedo a un rol específico
Entonces veré el listado de miembros que tienen ese rol asignado

Añadir miembro a un rol

Para poder tener una gestión correcta de los roles del sistema
Como administrador
Necesito añadir un miembro a un rol específico

Hay que tener en cuenta que un usuario no se le podrá añadir a un rol en el que ya sea miembro.

Escenario: Puedo añadir un miembro a un rol específico
Dado que soy un usuario con acceso a la aplicación
Cuando entro en el listado de roles
Y accedo a un rol específico
Y añado un miembro a ese rol
Entonces veré el listado de miembros con el miembro recién añadido.

Listar permisos de un rol

Para poder tener una gestión correcta de los roles del sistema
Como administrador
Necesito listar los permisos que tiene un determinado rol

Escenario: Puedo listar los permisos de un rol específico
Dado que soy un usuario con acceso a la aplicación
Cuando entro en el listado de roles
Y accedo a un rol específico
Entonces veré el listado de permisos que tiene ese rol específico

Editar permisos de un rol

Para poder tener una gestión correcta de los roles del sistema

Como administrador
Necesito editar permisos de un determinado rol

Escenario: Puedo editar los permisos de un rol específico
Dado que soy un usuario con acceso a la aplicación
Cuando entro en el listado de roles
Y accedo a un rol específico
Y actualizo un permiso de un rol
Entonces veré el listado de permisos actualizados que tiene ese rol específico

Añadir rol

Para poder tener una gestión correcta de los roles del sistema
Como administrador
Necesito poder añadir un rol con un nombre determinado

Escenario: Puedo añadir un rol
Dado que soy un usuario con acceso a la aplicación
Cuando añado un nuevo rol
Entonces veré en el listado de roles el nuevo rol creado

Borrar rol

Para poder tener una gestión correcta de los roles del sistema
Como administrador
Necesito borrar un determinado rol

Escenario: Puedo borrar un rol
Dado que soy un usuario con acceso a la aplicación
Cuando borro un rol ya existente
Entonces no veré en el listado de roles el rol borrado

5.2.5. Calendario

Ver calendario

Para poder ver de forma rápida y de una forma global los eventos ocurridos en el proyecto
Como usuario
Necesito ver un calendario con los eventos

Escenario: Puedo ver los eventos en el calendario
Dado que soy un usuario con acceso a la aplicación
Y soy un estudiante con un proyecto asignado
Cuando accedo al calendario
Entonces veré el listado de eventos que han ocurrido.

5.2.6. Tienda

Listar productos a la venta

Para poder ver qué productos están disponibles para comprar
Como usuario
Necesito ver un listado de productos

Estos productos tendrán:

- **Coste por unidad (en Euros).**
- **Unidades disponibles.**

Se asignan automáticamente al proyecto que tenga la persona que lo ha comprado. Se añadirá en el proyecto como Adquisición.

Productos que pueden estar disponible:

- **Alquiler de local**
- **Internet**
- **Luz + Agua**
- **Ordenador**
- **Material de oficina**
- **Sillas**
- **Mesas**
- **Cables**
- **Ordenador Portátil**
- **Teclado. Ratón, Monitor**
- **Cajonera**

Escenario: Puedo listar los productos de la tienda
Dado que soy un usuario con acceso a la aplicación
Cuando accedo al listado de productos
Entonces veré el listado de productos con su stock.

Comprar producto

Para poder realizar los proyectos de una forma adecuada
Como usuario
Necesito poder comprar los productos que necesite

Los productos se añadirán a la lista de compras del proyecto. El coste total debe ser menor que la cantidad de dinero que quede en el proyecto.

La cantidad mínima de producto que se puede comprar es 1.

La cantidad máxima de producto que se puede comprar es el stock de la tienda.

Escenario: Puedo comprar un producto que no exceda de la cantidad total del proyecto
Dado que soy un usuario con acceso a la aplicación
Y soy un estudiante con un proyecto asignado
Cuando accedo al listado de productos
Y compro un producto
Entonces veré el listado de productos con una unidad de stock menos.

Escenario: No puedo comprar un producto que sea mayor de la cantidad de dinero de presupuesto del proyecto
Dado que soy un usuario con acceso a la aplicación
Y soy un estudiante con un proyecto asignado
Cuando accedo al listado de productos
Y compro un producto que exceda del presupuesto
Entonces veré un mensaje de error de la aplicación.

5.2.7. Gestión de proyectos

Listar miembros

Para poder ver qué personas pertenecen a un proyecto
Como usuario
Necesito ver un listado de los miembros de ese proyecto

Solo podrán ver los miembros de un proyecto una persona que:
- **Sea miembro de ese proyecto**
- **Sea administrador**

Escenario: Puedo listar los miembros de mi proyecto
Dado que soy un usuario con un proyecto asignado
Cuando entro en el proyecto asignado
Entonces veré el listado de miembros

Añadir miembro a un proyecto

Para poder gestionar de forma correcta un proyecto
Como administrador
Necesito poder añadir miembros a un proyecto

No se puede añadir un miembro que ya está dentro del proyecto

Escenario: Puedo añadir un miembro a un proyecto
Dado que soy un usuario con acceso a la aplicación
Cuando entro en un proyecto
Y añado un miembro a un proyecto
Entonces veré en el listado de miembros al nuevo miembro añadido

Listar productos asignados a un proyecto

Para poder gestionar de forma correcta un proyecto
Como usuario
Necesito poder listar los productos que tiene ese proyecto

Escenario: Puedo listar los productos adquiridos para un proyecto
Dado que soy un usuario con acceso a la aplicación
Cuando entro en un proyecto
Entonces veré el listado de productos asignados al proyecto

Listar eventos producidos en un proyecto

Para poder conocer qué ocurre en el día a día de un proyecto
Como usuario
Necesito poder listar los eventos que se producen en un proyecto

Habrá eventos que sumarán gastos al proyecto.

Un ejemplo de evento sería:
- **Un miembro se ha unido al proyecto.**
- **Un miembro ha sido eliminado del proyecto.**
- **Un miembro del proyecto está enfermo, por lo tanto, permanece en casa descansando (ese miembro no puede imputar horas).**
- **Un portátil se ha averiado (si no hay portátiles extra, ese miembro no puede trabajar).**

Escenario: Puedo listar eventos de un proyecto
Dado que soy un usuario con acceso a la aplicación

Cuando entro en un proyecto
Entonces veré el listado de eventos que han ocurrido en el proyecto

Ver información del proyecto

Para poder tener una visión global del proyecto
Como usuario
Necesito poder ver información básica del proyecto

Un listado de datos básicos serían:

- **Nombre del proyecto**
- **Fecha de creación**
- **Presupuesto del proyecto**
- **Cantidad del presupuesto disponible**
- **Desglose del presupuesto**

Escenario: Puedo ver la información de un proyecto
Dado que soy un usuario con acceso a la aplicación
Cuando entro en un proyecto
Entonces veré la información del proyecto

5.2.8. Gestión de los documentos del proyecto

Listar documentos de un proyecto

Para poder tener una visión de los documentos asociados al proyecto
Como usuario
Necesito poder listar los documentos de ese proyecto

Escenario: Puedo listar documentos de un proyecto
Dado que soy un usuario con acceso a la aplicación
Cuando entro en un proyecto
Y accedo al listado de documentos
Entonces veré el listado de documentos

Añadir un documento al proyecto

Para poder gestionar de forma correcta los documentos de un proyecto
Como usuario

Necesito poder añadir un documento al proyecto

Escenario: Puedo añadir un documento a un proyecto
Dado que soy un usuario con acceso a la aplicación
Cuando entro en un proyecto
Y añado un documento a un proyecto
Entonces veré en el listado de documentos el nuevo documento añadido

Editar un documento al proyecto

Para poder gestionar de forma correcta los documentos de un proyecto
Como usuario
Necesito poder editar un documento al proyecto

Escenario: Puedo editar un documento ya existente de un proyecto
Dado que soy un usuario con acceso a la aplicación
Cuando entro en un proyecto
Y edito un documento a un proyecto
Entonces veré en el listado de documentos el documento editado

Borrar un documento al proyecto

Para poder gestionar de forma correcta los documentos de un proyecto
Como usuario
Necesito poder borrar un documento al proyecto

Escenario: Puedo borrar un documento ya existente de un proyecto
Dado que soy un usuario con acceso a la aplicación
Cuando entro en un proyecto
Y borro un documento a un proyecto
Entonces dejaré de ver en el listado de documentos el documento borrado

5.2.9. Documentación

Ver documentación

Para poder usar de forma correcta la aplicación
Como usuario
Necesito poder acceder a la documentación

La documentación estará compuesta por distintas secciones:

- **Sección X**
- **Sección Y**
- **Sección Z**

Escenario: Puedo ver la documentación
Dado que soy un usuario dentro de la aplicación
Cuando voy a ver la documentación
Entonces veré la documentación compuesta por distintas secciones.

5.3. Estudio de alternativas a la solución

A continuación se listan diferentes alternativas a la solución.

5.3.1. The Project Management Game

Este juego es para un solo jugador. Es vía web y está basado en asignar a diferentes miembros del equipo una serie de tareas.

Cada miembro del equipo tiene una característica. Es decir, puede ser un trabajador más rápido/lento, más caro/barato o simplemente un trabajador normal. Tienes un listado de tareas y un presupuesto asignado. Lo importante es:

- Entregar a tiempo el proyecto.
- El coste del proyecto sea menor al presupuesto.

Se han realizado distintos proyectos con tareas distintas. Cuando el proyecto se termina salen las conclusiones:

- Coste por cada empleado.
- Porcentaje de completado de cada tarea.

La imagen siguiente muestra un ejemplo de un proyecto que se ha entregado a tiempo y por debajo del presupuesto.

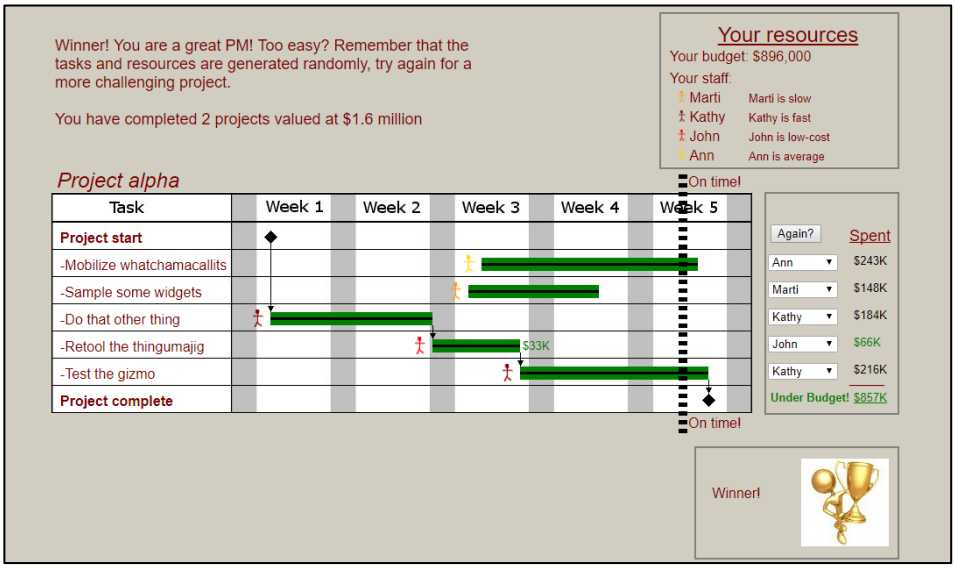

Ilustración 2 - Visualización de "The Project Management Game"

Los inconvenientes de este juego son los siguientes:

- No permite tener un equipo de estudiantes trabajando en un mismo proyecto.
- El enfoque es simplista en el aspecto de que se basa en completar tareas (algunas con dependencias) y el gasto del presupuesto. No tiene la posibilidad de que existan eventos ni tener que comprar material para el proyecto.

5.3.2. Sharkworld

Sharkworld [2] es un juego online basado en la gestión de un proyecto basado en el mundo de los tiburones. El juego permite distintas acciones sobre distintos módulos:

- Presupuesto. Visión general del presupuesto.
- Personajes: Se presentarán mientras se está jugando y tienen distintos roles. Se pueden mantener conversaciones y se pueden conseguir una bonificación si se pregunta de forma correcta.
- Lugares: Se puede viajar a distintas zonas como el Cloud9 bar, apartamento, etc.
- Correo electrónico: Recibir correos en el portátil.
- Planificar tareas: Las tareas llevan un tiempo poder completarlas por lo que se plantea un diagrama donde se pueden ir realizando.

Es un simulador bastante completo para proyectos software con una temática sobre tiburones. El precio del juego para una persona es de 150€ (más impuestos).

5.3.3. Implementación y desarrollo a medida de la solución

Otra solución alternativa sería implementar y desarrollar una solución que sea acorde a los requisitos. Esta solución puede implementarse utilizando distintos lenguajes de programación (Java, Python, Ruby, PHP, etc).

Las ventajas que presenta realizar el desarrollo de la solución son:

- Libertad. Utilizar una plataforma externa suele llevar a que la aplicación esté sujeta a numerosas políticas y que limiten nuestro control sobre la aplicación.

- Competencia. Usar una plataforma propia es un factor de diferenciación ya que el resto de competidores puede que usen la misma plataforma externa con las limitaciones que ello conlleva.

- Calidad. Normalmente usar una plataforma externa proporciona una imagen genérica (y normalmente de bajo coste), por lo que si se realiza de forma propia puede dar un aspecto más profesional.

- Control. El mundo es cambiante y hay momentos en la vida en la que surgen nuevas necesidades de negocio. Por ello, tener una solución propia tiene la posibilidad de poder alterarla (enfocándose en otro aspecto).

- Experiencia como desarrollador. He trabajado como desarrollador en distintas empresas y tengo grandes conocimientos técnicos sobre cómo dar una solución para un problema técnico.

Las desventajas que existen al tener una solución propia son:

- Conocimiento. Normalmente el conocimiento es dependiente de los creadores de la plataforma. A diferencia de un sistema externo que implica que existe un mayor número de desarrolladores. Además de que suele existir una comunidad que ayuda en los problemas que puedas tener.

- Coste. El coste de implementar una solución a medida suele ser mayor que utilizar una aplicación externa.

5.4. Justificación de la elección de la solución

Para la realización de este Trabajo de Fin de Grado, se ha decidido por la solución planteada en el apartado 5.3.3. Implementación y desarrollo a medida de la solución. Esta decisión está basada en los siguientes motivos:

- Crear una aplicación con total libertad. Las aplicaciones de terceros no tienen varias funcionalidades necesarias para el problema que se busca encontrar una solución. Por ejemplo, es importante que los estudiantes puedan formar equipo y poder trabajar en equipo es uno de los valores que se quieren transmitir al alumnado.

- Aplicación de mayor calidad. Al haber trabajado como desarrollador web, he desarrollado distintas aplicaciones y puedo aplicar mis conocimientos. Además de poder entregar más funcionalidades en menos tiempo. Además de eliminar gran parte de las incertidumbres.

- Mejorar como desarrollador de aplicaciones web. Es importante como ingeniero informático poder aplicar los conocimientos y crear una solución que cubra todos los requisitos del usuario.

- Desarrollo de una aplicación en la que añadir funcionalidades no sea un problema. La aplicación debe poder añadir o modificar funcionalidades existentes de manera que se puedan adaptar a una mejora en el aprendizaje en el alumno.

Capítulo 6: Diseño del sistema

En este capítulo se define la arquitectura software que compone el sistema enfocada a resolver las funcionalidades del sistema. La arquitectura software se va a componer de los siguientes elementos:

- Componentes
- Módulos
- Datos

6.1. Definición de la arquitectura del sistema

La arquitectura del sistema está basada en distintos aspectos que se van a detallar en las siguientes secciones.

6.1.1. Arquitectura técnica

La arquitectura técnica de la aplicación se basa en distintas capas:

- Capa de datos o capa de persistencia: En esta capa se encuentra la base de datos y las tablas que la componen (junto con los datos).
- Capa de aplicación: Esta capa está dividida en dos módulos:
    - Backend: En este módulo está la lógica de negocio y se procesan todas las consultas a la base de datos.
    - Frontend: Este módulo se visualiza la aplicación web.
- Capa de servicio: Esta capa dirige las peticiones del usuario para que pueda acceder a la aplicación.

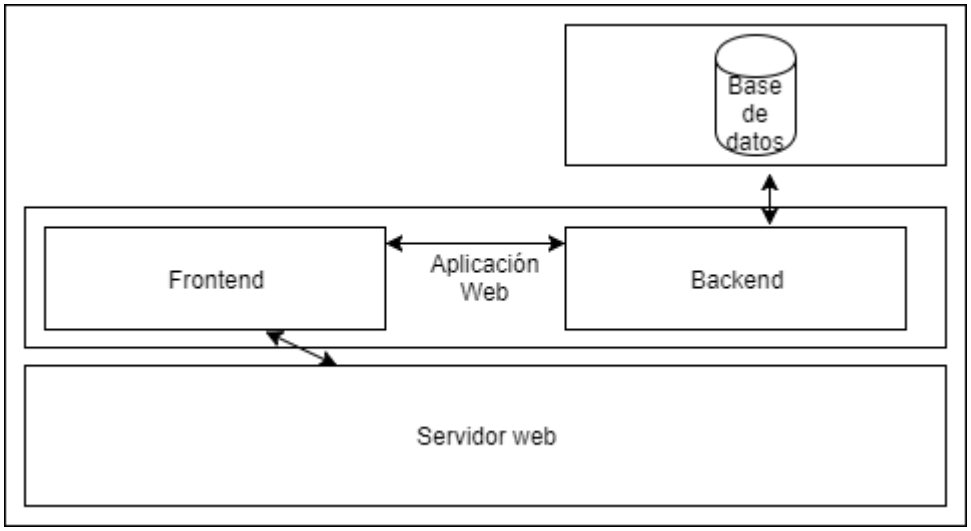

Ilustración 3 - Arquitectura técnica

Al utilizarse Laravel permite usar dependiendo de cómo se configure el tipo de base de datos a la cual va a consultar.

El archivo de entorno configura la base de datos y los datos de acceso:

Contenido del archivo ".env"
[…] **DB_CONNECTION=mysql** **DB_HOST=127.0.0.1** **DB_DATABASE=simulatordps** **DB_USERNAME=root** **DB_PASSWORD=** **[…]**

Tabla 7 - Contenido del archivo ".env"

En este caso, la base de datos es MySQL y se encuentra alojada en el propio equipo. Para crear las tablas y rellenar con los datos necesarios con ejecutar el siguiente comando automática se realizaría esta operación: "php artisan migrate:refresh --seed".

6.1.2. Patrón Modelo-Vista-Controlador

Laravel usa el patrón Modelo-Vista-Controlador (MVC). En este caso, el usuario al acceder a la aplicación y realizar una acción, el controlador tendrá la lógica de negocio y realizará las consultas a la base de datos que sean necesarias. Cuando el controlador recibe la respuesta de los modelos, podrá devolver los resultados en una vista.

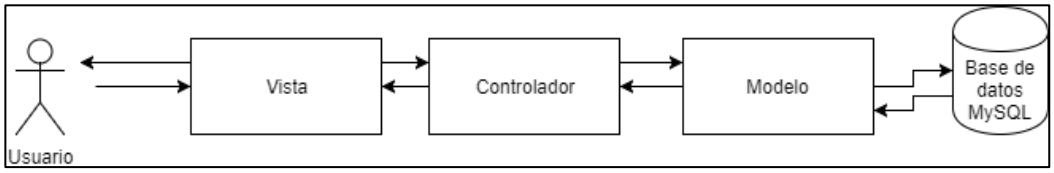

Ilustración 4 - Modelo Vista Controlador

Laravel para la generación de vistas usa las plantillas Blade. Estas plantillas tienen las siguientes características:

- Permite herencia entre plantillas. Se puede definir una plantilla "maestra" de las cuales el resto pueden heredar todo lo necesario menos el contenido que es distinto en cada una de ellas.
- Permite tener componentes aislados. Gracias al uso de la directiva "@component" es posible inyectar componentes cuando sea necesario su uso.
- Tiene estructuras de control. Hay estructuras de control que se pueden usar de manera más elegante (sin ser código). Esto hace que el código sea más fácil de entender.

Para las consultas a la base de datos, Laravel utiliza Eloquent ORM. Eloquent ORM es una implementación básica de ActiveRecord. Por ejemplo, para guardar un resultado sería de la siguiente manera:

```
$flight = new Flight;

$flight->name = $request->name;

$flight->save();
```

Tabla 8 - Modelos haciendo uso de Eloquent ORM

6.1.3. Control de versiones

El control de versiones beneficia a la gestión de los diversos cambios sobre los elementos de la aplicación web. En este caso, se ha utilizado un repositorio Git (Bitbucket) ya que tiene flexibilidad y permite poder realizar cambios en local (además de poder usar ramas secundarias para desarrollar).

El sistema que se ha utilizado ha sido el siguiente:

- Utilizar la rama "master" como la rama de producción (la última desplegada).
- Utilizar las ramas "features/xxx" como las ramas para añadir funcionalidades a la rama de producción.

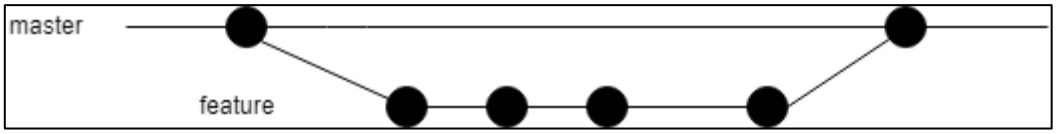

Ilustración 5 - Uso de rama feature y master

6.1.4. Integración Continua

La Integración Continua (CI) es una propuesta creada inicialmente por Martin Fowler que consiste en integrar de forma continua y de forma automática el código de un proyecto lo antes posible para así poder detectar fallos lo más rápido posible. Esto significa que el proyecto debe compilarse y lanzar todas las pruebas.

Para esta propuesta existen distintas herramientas disponibles:

- Jenkins [7]: Es la herramienta que será utilizada para realizar Integración Continua en este proyecto ya que es fácil de usar, ya se ha aplicado a otros proyectos que he desarrollado y se integra perfectamente con Git.
- CircleCI.
- Bamboo.
- Travis.

Jenkins [7] es un software de Integración Continua open source escrito en Java. Está basado en el proyecto Hudson.

Ilustración 6 - Logo de Jenkins

Para este proyecto se ha creado la siguiente tarea automática (se ejecuta cada 5 minutos en caso de que encuentre un cambio):

- Descargar código fuente de la rama “master” del repositorio Git.
- Actualiza las dependencias (en caso de que haga falta).
- Actualiza la base de datos (en caso de que haga falta).
- Si se produce un error, envía un mensaje de error.

Una visualización del proyecto de Jenkins sería esta:

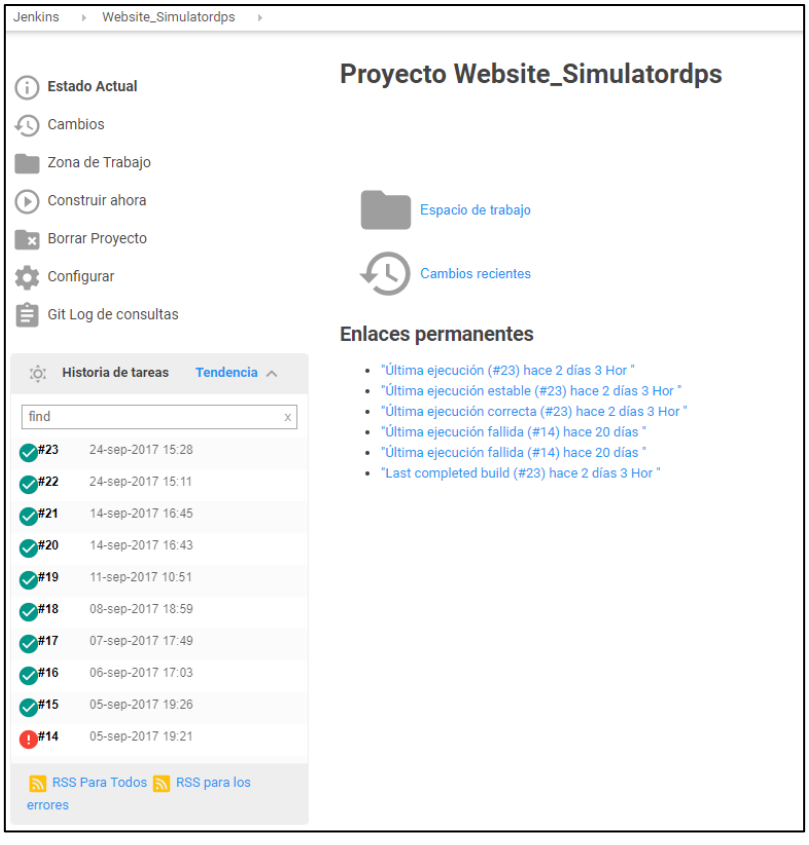

Ilustración 7 - Visualización proyecto Jenkins

6.3. Diseño de clases

Para el diseño de clases se va a basar en como lo dicta Laravel con la siguiente estructura de archivos:

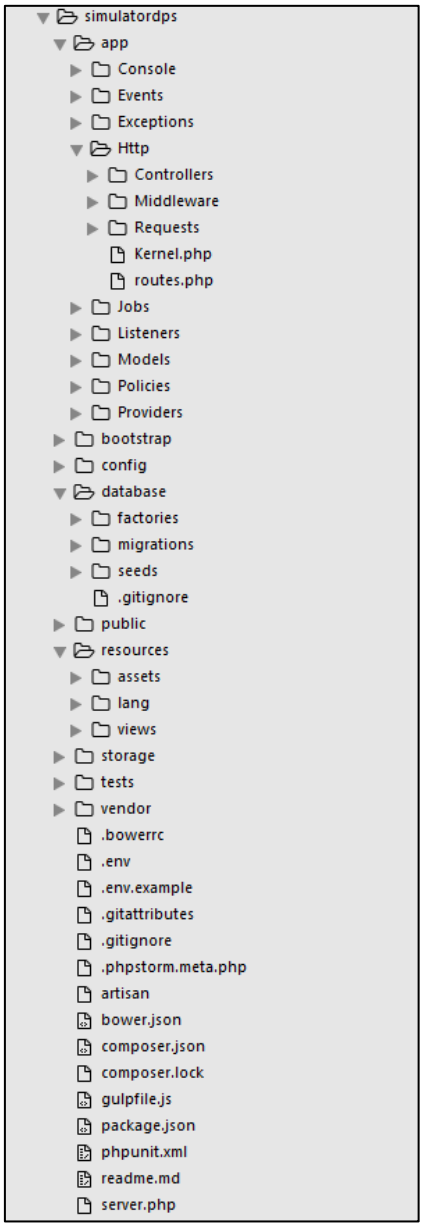

Ilustración 8 - Estructura de clases con Laravel

Además Laravel define la forma de crear las clases gracias a los comandos que provee “artisan”.

Las clases dedicadas a la lógica de negocio son las que se encuentran en: “app/Http/Controllers”.

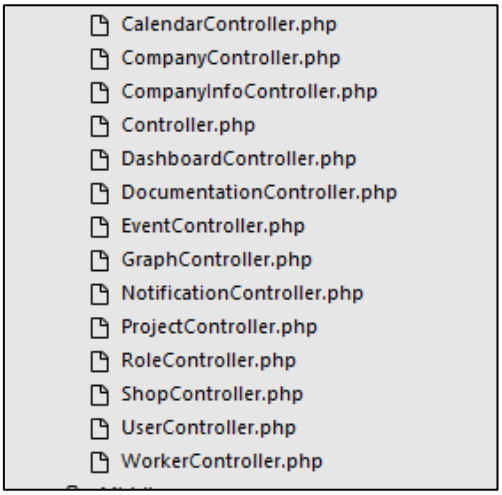

Ilustración 9 - Controladores

6.4. Especificaciones de excepciones

En la lógica de negocio se validan todas las llamadas y han sido probadas de forma exhaustiva para que el usuario no reciba ninguna excepción sin controlar. En el caso de que se produjera una excepción, Laravel recoge esa excepción, guarda en los logs la excepción que se produce y se envía un correo con la excepción producida.

6.5. Entorno tecnológico

El entorno tecnológico va a variar dependiendo de los requisitos que se han listado en el apartado 5.2. Funcionalidades del sistema. Para este caso práctico se va a utilizar el siguiente entorno tecnológico:

- Sistema operativo: Linux (recomendable distribución Ubuntu, Debian o CentOS).
- Servidor web: Apache Tomcat (es necesario definir un virtual host para poder acceder a la aplicación).
- Base de datos: MySQL (aunque también es posible utilizar PostgreSQL o SQLite).
- Disco duro: Es recomendable que sea un disco duro SSD (y con un tamaño de al menos 5 GB).

- Firewall: Es necesario para evitar ataques que exista un firewall (iptables). Además es recomendable “fail2ban” para prevenir de intrusos en el servidor.

- Copias de seguridad: Sería recomendable tener copias de seguridad de forma semanal. Si la copia de seguridad es incremental permite ahorrar mucho espacio de disco duro.

- Monitorización: Añadir reglas para que se reciban correos electrónicos si el disco duro está apunto de llenarse, si la base de datos está caída o si el disco duro del servidor web está saturado es muy recomendable.

6.6. Modelo de base de base de datos

A continuación se muestra el modelo de base de datos con las siguientes entidades:

- Usuarios
- Proyectos
- Artículos de la tienda
- Roles
- Permisos
- Control de acceso
- Documentos del proyecto
- Migraciones

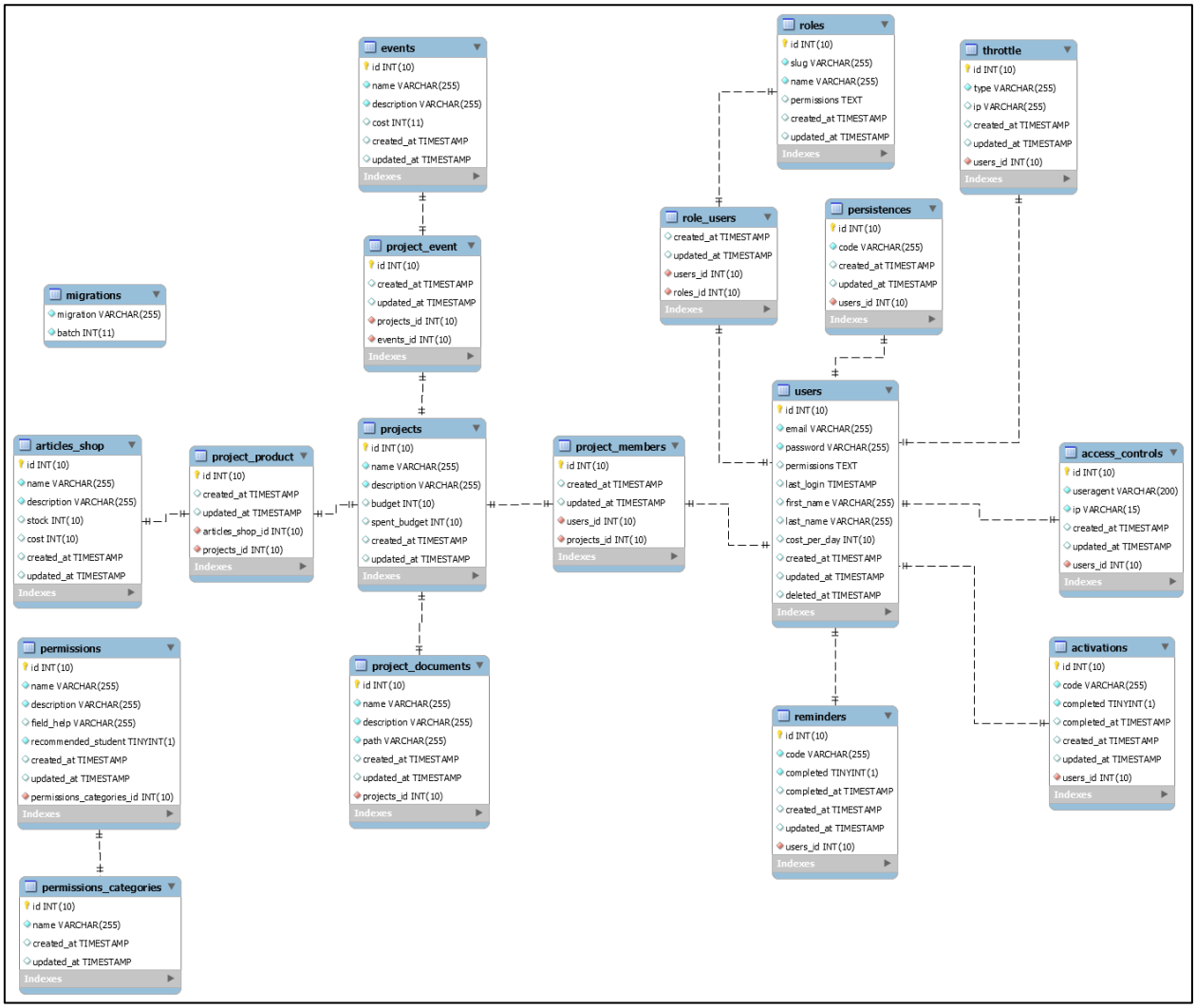

Ilustración 10 - Modelo de base de datos

Capítulo 7: Pruebas del sistema y manual de usuario

A continuación se van a listar todas las pruebas que se han realizado sobre el sistema y el manual de usuario. El manual de usuario provee una información completa del uso de la aplicación.

7.1. Definición de pruebas

Gracias al uso de "Behaviour-driven Development (BDD)" las funcionalidades del sistema cuentan con un listado de escenarios. Estos escenarios son, de facto, las pruebas que validan que el sistema funciona correctamente.

ID	Módulo	Funcionalidad	Escenario
001	Sesión del usuario	Iniciar sesión	Puedo hacer login con un usuario válido
002	Sesión del usuario	Iniciar sesión	No puedo hacer login con un usuario incorrecto
003	Sesión del usuario	Iniciar sesión	No puedo hacer login si no introduzco el correo
004	Sesión del usuario	Iniciar sesión	No puedo hacer login si no introduzco la contraseña
005	Sesión del usuario	Iniciar sesión	No puedo hacer login si introduzco un email sin formato correcto
006	Sesión del usuario	Cerrar sesión	Puedo cerrar sesión con un usuario correcto
007	Sesión del usuario	Duración de la sesión	Mi sesión se cierra al pasar una duración de tiempo predeterminada
008	Sesión del usuario	Una sesión por navegador	Mi sesión sólo estará abierta en un navegador
009	Cuenta del usuario	Cambiar contraseña	Puedo cambiar de contraseña por una distinta
010	Gestión de usuarios	Ver cuentas de los usuarios	Puedo listar los usuarios existentes
011	Gestión de usuarios	Ver detalle de una cuenta de usuario	Puedo ver los detalles de un usuario
012	Gestión de usuarios	Editar cuenta de usuario	Puedo editar una cuenta de usuario
013	Gestión de usuarios	Añadir cuenta de usuario	Puedo añadir una cuenta de usuario
014	Gestión de usuarios	Borrar cuenta de usuario	Puedo borrar una cuenta de usuario que no sea la cuenta propia

015	Gestión de roles	Ver roles	Puedo listar los roles existentes en la aplicación
016	Gestión de roles	Ver miembros de un rol	Puedo listar los miembros que tengan un rol un asignado
017	Gestión de roles	Añadir miembro a un rol	Puedo añadir un miembro a un rol específico
018	Gestión de roles	Listar permisos de un rol	Puedo listar los permisos de un rol específico
019	Gestión de roles	Editar permisos de un rol	Puedo editar los permisos de un rol específico
020	Gestión de roles	Añadir rol	Puedo añadir un rol
021	Gestión de roles	Borrar rol	Puedo borrar un rol
022	Calendario	Ver calendario	Puedo ver los eventos en el calendario
023	Tienda	Listar productos a la venta	Puedo listar los productos de la tienda
024	Tienda	Comprar producto	Puedo comprar un producto que no exceda de la cantidad total del proyecto
025	Tienda	Comprar producto	No puedo comprar un producto que sea mayor de la cantidad de dinero de presupuesto del proyecto
026	Gestión de proyectos	Listar miembros	Puedo listar los miembros de mi proyecto
027	Gestión de proyectos	Añadir miembro a un proyecto	Puedo añadir un miembro a un proyecto
028	Gestión de proyectos	Listar productos asignados a un proyecto	Puedo listar los productos adquiridos para un proyecto
029	Gestión de proyectos	Listar eventos producidos en un proyecto	Puedo listar eventos de un proyecto
030	Gestión de proyectos	Ver información del proyecto	Puedo ver la información de un proyecto
031	Gestión de los documentos del proyecto	Listar documentos de un proyecto	Puedo listar documentos de un proyecto
032	Gestión de los documentos del proyecto	Añadir un documento al proyecto	Puedo añadir un documento a un proyecto
033	Gestión de los documentos del proyecto	Editar un documento al proyecto	Puedo editar un documento ya existente de un proyecto
034	Gestión de los documentos del proyecto	Borrar un documento al proyecto	Puedo borrar un documento ya existente de un proyecto

035	Documentación	Ver documentación	Puedo ver la documentación

Tabla 9 - Tabla con Pruebas de funcionalidades

Estas pruebas cubren el 100% de la funcionalidad que se han solicitado ya que contienen todos los escenarios que se han implementado. Esto se debe a la gran ventaja que proporciona el uso de BDD.

7.2. Trazabilidad de las pruebas con requisitos

En este Trabajo de Fin de Grado, al utilizar BDD crear una matriz de trazabilidad de las pruebas con los requisitos no aporta mucho valor ya que:

- Cada funcionalidad es un requisito.
- Cada prueba es un escenario que está dentro de esa funcionalidad.

7.3. Resultados de las pruebas realizadas

Las pruebas que se han realizado han sido las siguientes:

Fecha	Número de pruebas	Fallidas	Correctas	Porcentaje	Estado
01/08/2017	20	4	16	80%	KO
15/08/2017	35	35	35	100%	OK
27/08/2017	35	35	35	100%	OK

Tabla 10 - Tabla con los resultados de las pruebas

Se han ejecutado las pruebas del sistema tres veces:

- La primera ejecución no fue completa ya que se detectaron pruebas que fallaron y se tenía que arreglar varias funcionalidades.
- La segunda y tercera ejecución de las pruebas han sido satisfactorias (estado OK). Es decir, todas las pruebas se pasaron y se verificaron todas las funcionalidades de la aplicación.
- Se han ejecutado en distintos navegadores para validar que funciona correctamente en distintas plataformas. Los navegadores que se han probado han sido los siguientes: Mozilla Firefox, Google Chrome e Internet Explorer.

7.4. Manual de usuario

En este apartado se hablará de los módulos y las funcionalidades que se pueden utilizar en la plataforma.

7.4.1. Acceso a la aplicación

Para poder acceder a la aplicación es necesario que tengas una cuenta de usuario en el sistema. En el caso de que no la tengas, debes solicitar una cuenta de usuario al administrador del sistema.

Los datos necesarios para solicitar serían los siguientes:

- Correo electrónico.
- Nombre.
- Apellidos.

La contraseña será generada de forma aleatoria, por lo que no hará falta incluirla en la solicitud de creación de nueva cuenta.

Para iniciar sesión en la aplicación, basta con acceder a la página principal. En caso de que no hayamos hecho login, aparecerá la siguiente pantalla.

Ilustración 11 - Acceso a la aplicación

Los datos que debe introducir el usuario para poder acceder a la aplicación son los siguientes:

- Correo electrónico.
- Contraseña asignada.

7.4.2. Visión general

Al acceder de forma correcta a la aplicación, podremos ver una visión general con datos de interés. Esta vista dependerá del rol del usuario. Si el rol es de administrador verá el siguiente resumen:

- Número de estudiantes en la aplicación.
- Número de proyectos creados.
- Número de productos en la tienda.

Ilustración 12 - Visión general de la aplicación

Si el rol del usuario es de estudiante y tiene asignado un proyecto, podrá ver la información sobre su proyecto y poder acceder al proyecto.

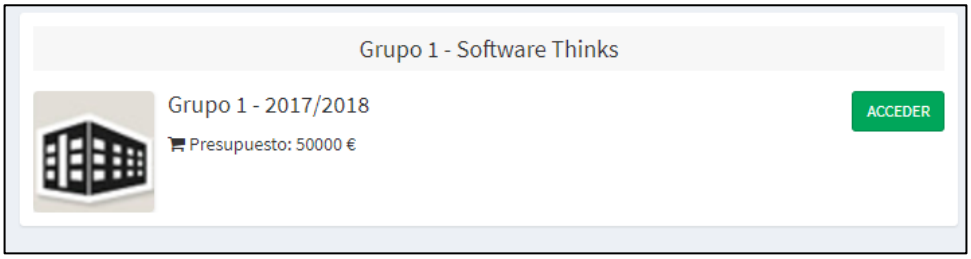

Ilustración 13 - Acceso al proyecto asignado

En el caso de que el usuario sea estudiante pero todavía no se le haya asignado un proyecto, recibirá el siguiente aviso.

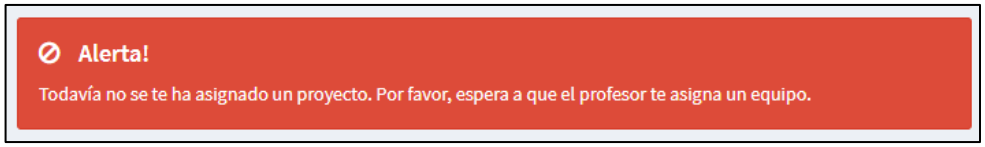

Ilustración 14 - Alerta al no tener asignado un proyecto

Este usuario no podrá realizar compras en la tienda pero si podrá leer la documentación de la aplicación.

7.4.3. Proyectos

El administrador podrá crear proyectos en la pestaña de “Proyectos” > “Añadir proyecto”. Para poder crear un proyecto deberá introducir los siguientes datos:

- Nombre.
- Descripción.
- Presupuesto inicial.

El administrador también podrá visualizar los proyectos. Deberá ir a la pestaña “Proyectos” > “Gestión de proyectos”. Un ejemplo de cómo se listarían los proyectos sería la siguiente imagen:

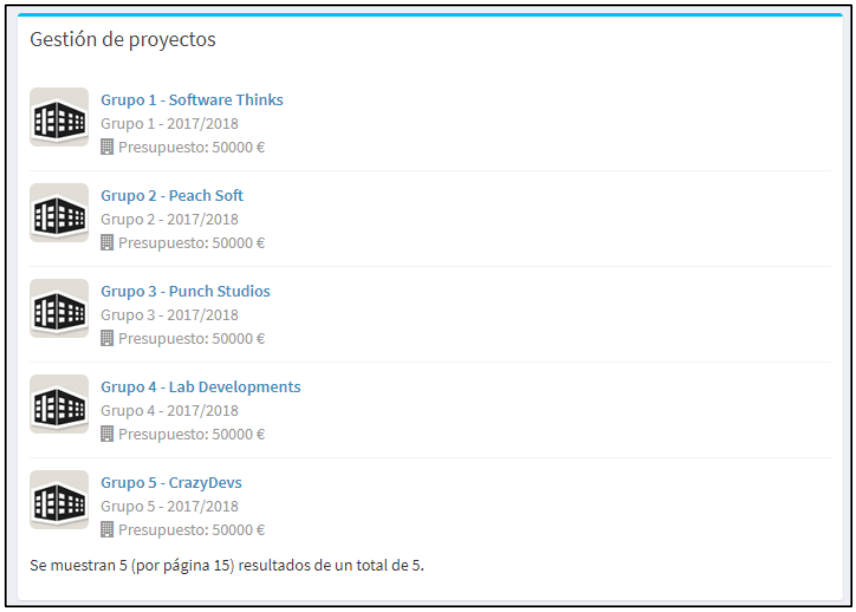

Ilustración 15 - Listado de proyectos

Al visualizar un proyecto, se mostrará:

- Opciones para la gestión de elementos relacionados con el proyecto.
- Resumen del presupuesto.
- Inicio del proyecto, eventos, productos comprados, etc que ha ocurrido en el proyecto.

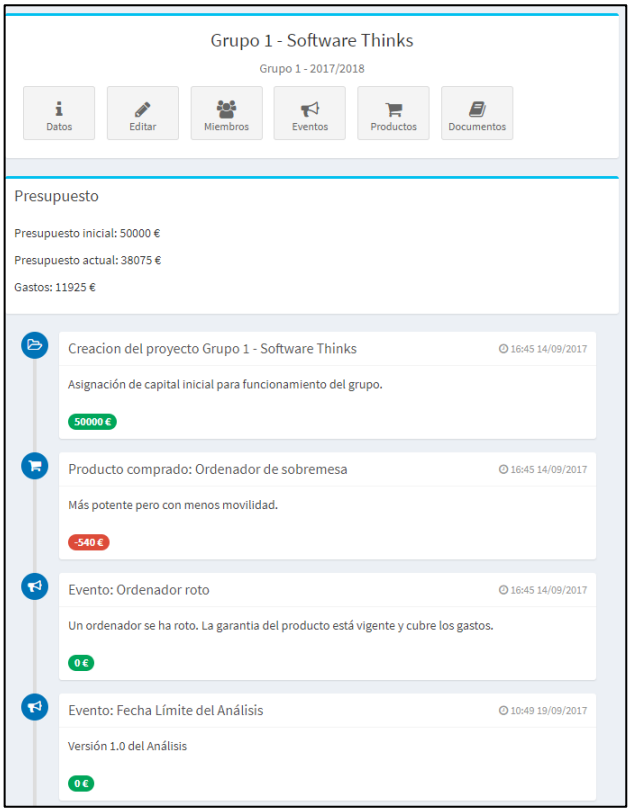

Ilustración 16 - Visualización de proyecto

En esta vista se podrá realizar las siguientes acciones:

- Listar miembros.
- Listar eventos.
- Listar productos.
- Listar documentos.
- Añadir, editar y borrar documentos.

Las acciones que solo podrán ejecutar los administradores:

- Editar proyecto.
- Añadir miembros.
- Añadir eventos.

7.4.4. Tienda

La tienda estará disponible para todos los usuarios. Siendo el administrador el único que podrá añadir productos. El estudiante que tenga un proyecto podrá comprar los productos que necesite para su proyecto.

Los artículos que compre harán disminuir el presupuesto del proyecto y listarán como productos asignados a ese proyecto.

Los productos en la tienda tienen un número de stock. Esto va a llevar a que:

- Los miembros de los proyectos puedan competir en dejar sin stock a otros jugadores de otros equipos.
- El administrador podrá penalizar los proyectos que no tengan determinados proyectos.
- Al acabar el stock, nadie podrá comprar ese producto.

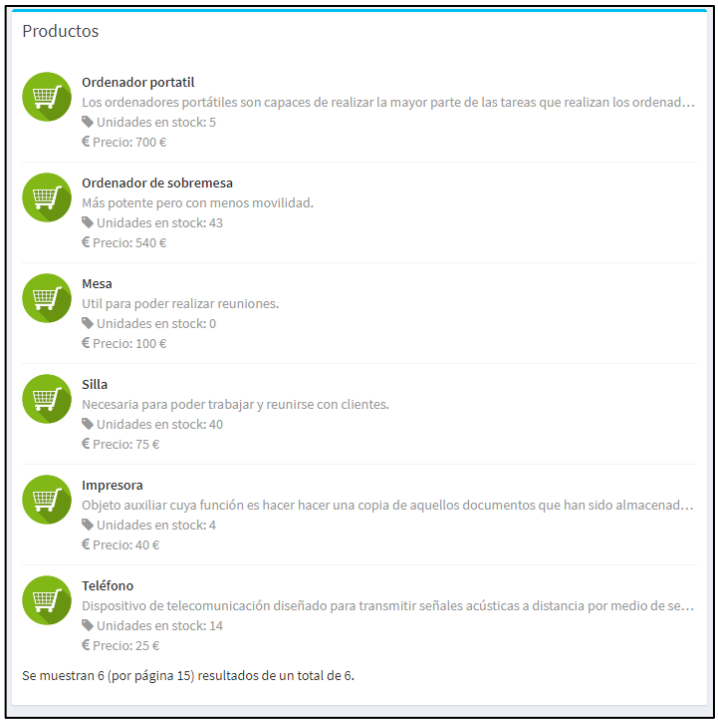

Ilustración 17 - Listado de productos en la Tienda

7.4.5. Eventos

Para poder visualizar y poder añadir eventos es necesario ser administrador de la aplicación. Los eventos pueden provocar un coste extra al presupuesto o impulso al mismo. Incluso pueden ser neutros, es decir, ni beneficio ni pérdida.

En la imagen se muestran ejemplos de los tres casos:

- Coste en un proyecto: El evento de “Oficina inundada” sería un ejemplo.
- Beneficio para el proyecto: En este caso sería el evento “Subvención a la empresa”.
- Evento neutro: Un evento neutro es que no ha sido un coste para el proyecto pero tampoco ha sido un beneficio. En este caso, el evento “Ordenador roto” sería un evento neutro.

Ilustración 18 - Listado de eventos

7.4.6. Calendario

El estudiante que tenga un proyecto asignado podrá ver los eventos para poder gestionar mejor su proyecto. Cada evento tendrá información útil.

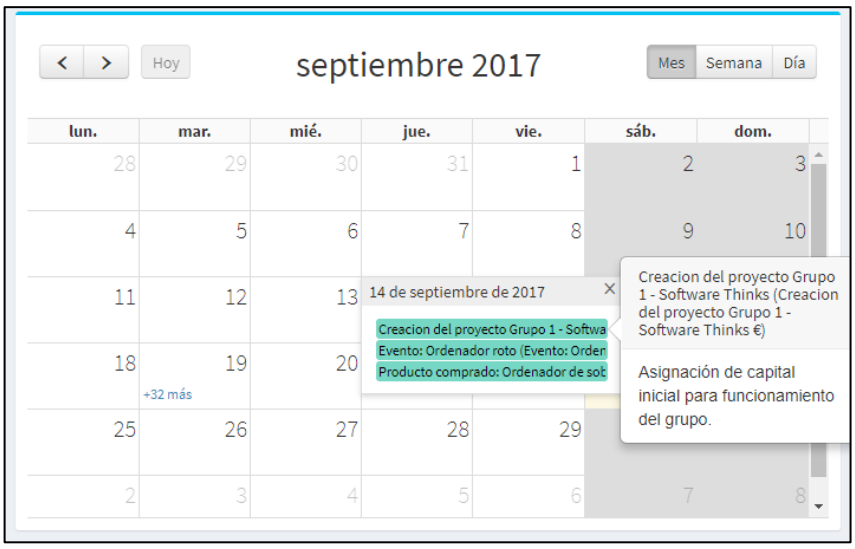

Ilustración 19 - Calendario con eventos del proyecto

7.4.7. Usuarios y roles

Esta sección solo tiene acceso el administrador de la aplicación y puede realizar las siguientes acciones:

- Listar qué usuarios han accedido a la aplicación (junto con intentos fallidos a la aplicación).
- Listar, añadir, editar y borrar usuarios.
- Listar, añadir, editar y borrar roles.
- Añadir o eliminar permisos a un determinado rol.
- Añadir o eliminar miembros a un determinado rol.

El sistema de permisos es por rol. Es decir, un usuario, por defecto, no tiene permiso para realizar ninguna acción en la aplicación. Si al usuario se le asigna un rol heredará todos los permisos que tiene ese rol.

El sistema es dinámico y se podría tener distintos roles para realizar acciones específicas. Un ejemplo de posibles roles podría ser:

- Moderador: Sería un administrador pero con menos permisos (no podría eliminar ningún elemento).
- Auditor: Sería un rol que podría visualizar la aplicación pero no podrá realizar ninguna acción en la misma.

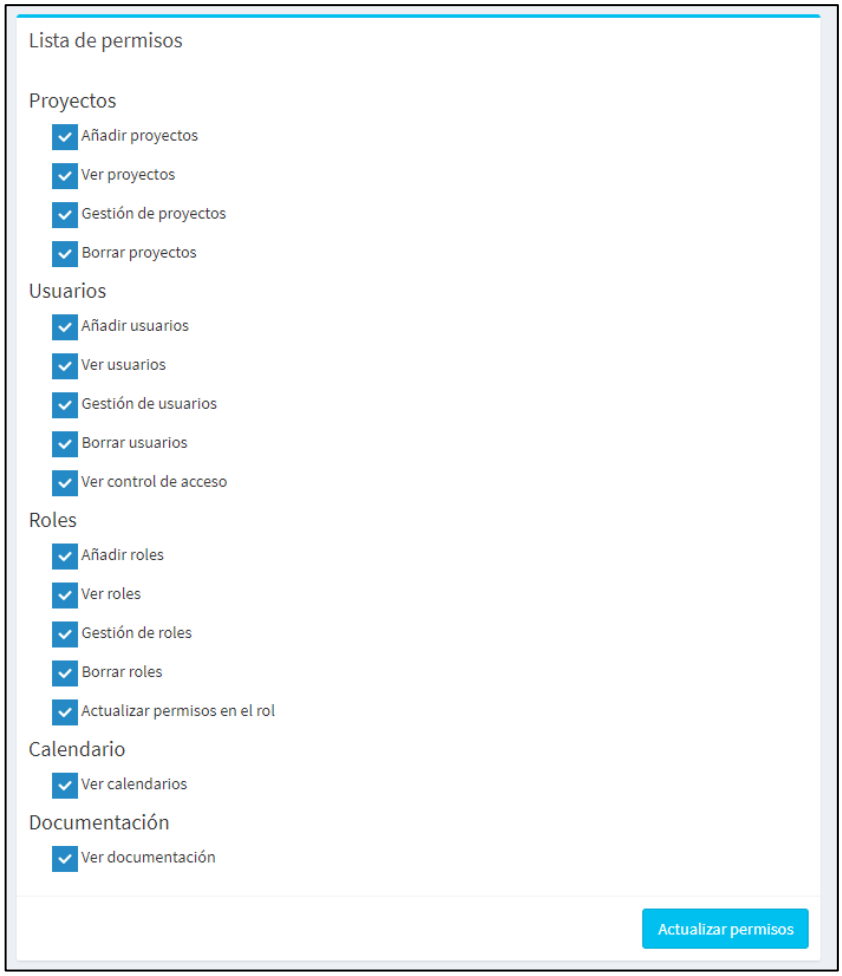

Ilustración 20 - Listado de permisos disponibles para un usuario

Por defecto, en la aplicación habrá dos roles:

- Administrador. Puede realizar todo tipo de acciones en la aplicación.
- Estudiante. Solo podrá realizar acciones dentro de su propio proyecto.

Capítulo 8: Planificación del proyecto

Uno de los apartados más importantes en la realización del Trabajo de Fin de Grado es la planificación. El presupuesto se ha realizado teniendo en cuenta este apartado ya que se realiza uso sobre distintos recursos.

8.1. Planificación

La planificación en este proyecto se realizará teniendo en cuenta las tres etapas en las que se va a dividir:

- Etapa de planificación.
- Etapa de desarrollo.
- Etapa de documentación.

8.1.1. Etapa de planificación

En esta etapa de planificación se va a determinar qué objetivos y qué funcionalidades se esperan del Trabajo de Fin de Grado. Además de incluir un marco temporal en la que el proyecto debe ser entregado. Las fases más importantes han sido las siguientes:

- Reunión para conocer el alcance del Trabajo de Fin de Grado. Se puede encontrar las funcionalidades que se van a desarrollar en la sección 5.2. Funcionalidades del sistema.
- Determinar un marco temporal en la que el proyecto va a ser entregado estimando por cada fase un tiempo determinado. Se ha realizado un diagrama de GANTT para tener una visión temporal del proyecto y de los hitos que se van completando en la sección 8.1. Planificación.
- Estudio de aplicaciones alternativas y las soluciones que han tomado para llevarlo a cabo. Este estudio ayuda a extraer qué funcionalidades aportar mayor valor al cliente y cuáles no.
- Estudio de las herramientas para desarrollar el Trabajo de Fin de Grado. Las herramientas que se han utilizado se encuentran en la sección 2.2.3. Herramientas utilizadas y han sido seleccionadas ya que aportan más valor para entregar el Trabajo de Fin de Grado.

8.1.2. Etapa de desarrollo

Esta etapa se va a llevar a cabo las funcionalidades que han sido solicitadas por el cliente. El desarrollo también estará complementado con las pruebas del sistema. Es decir, se va a validar todas las funcionalidades que se realicen para incrementar la calidad del software que se entrega. Las pruebas que se han realizado se encuentran en la sección 7.1. Definición de pruebas. Además la etapa de desarrollo se ha realizado las siguientes tareas:

- Desarrollo e implementación de la capa de negocio. La capa de negocio es donde se implementa qué se debe hacer en determinadas situaciones. Es decir, si un alumno es miembro de un proyecto podrá ver su proyecto.
- Desarrollo de la capa de persistencia de datos. Es necesario tener que almacenar los datos que sean necesarios para la gestión de proyectos.
- Diseño de la interfaz de usuario. La interfaz de usuario se realiza con HTML5, CSS y Javascript.

8.1.3. Etapa de documentación

En esta etapa se va a documentar todo el proyecto que se ha realizado. La aplicación se encuentra desarrollada y se evalúa todos los eventos que han ocurrido con ella. Además se añaden las conclusiones y las dificultades encontradas.

8.2. Diagrama de Gantt

El diagrama de Gantt será planificado por semanas (ver figura del Diagrama) y siendo:

- Fecha inicial: 30 de septiembre de 2016.
- Fecha finalización: 22 de septiembre de 2017.
- Duración total: Aproximadamente un año (358 días ó 51 semanas).

La estructura que se ha seguido para realizar este Trabajo de Fin de Grado ha sido una estructura simple y básica en la que poder entender qué hay que hacer para poder llegar a entregar el Trabajo de Fin de Grado. Siendo las siguientes etapas:

- Etapa de Planificación. Etapa inicial donde se va realizar varios estudios iniciales y se toman los requisitos necesarios.

- Etapa de desarrollo: Implementación de la solución para el Trabajo de Fin de Grado.

- Etapa de documentación: Creación de la memoria que será entregada con el Trabajo de Fin de Grado.

- Entrega de TFG: Es la etapa final en la que se realiza la entrega final y las gestiones asociadas a la misma.

Para realizar el Diagrama de Gantt se ha utilizado la herramienta “Tom’s Planner” [5]. Es una herramienta muy intuitiva y muy fácil de usar. Además tiene varios tutoriales que permiten a cualquier usuario poder crear un diagrama en cuestión de minutos.

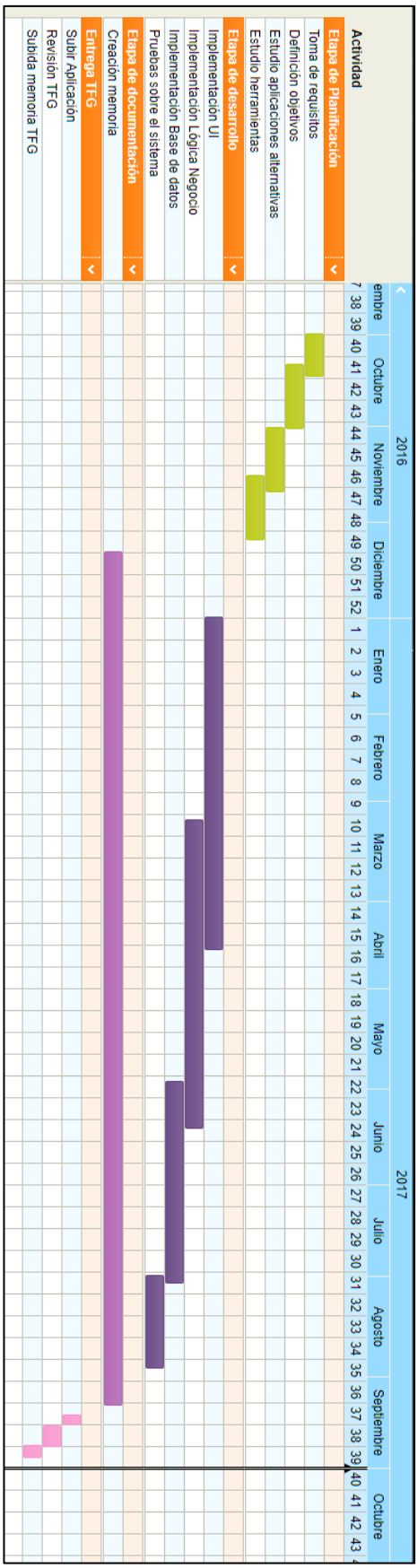

Ilustración 21 - Diagrama de Gantt

Capítulo 9: Conclusiones y futuras mejoras

Para realizar un análisis de todas las complicaciones que han existido al realizar el Trabajo de Fin de Grado se ha dividido en diferentes apartados. En toda realización de un proyecto siempre hay una serie de dificultades existentes (algunas se verán más a primera vista y otras durante la realización del mismo).

9.1. Dificultades encontradas

De forma resumida, me gustaría ser positivo y pensar que no he encontrado muchas dificultades. Aun así, no está de más, compartirlas con el resto de los lectores:

- **Realizar el Trabajo de Fin de Grado mientras se tiene una jornada laboral de 8 horas de lunes a viernes.** Es difícil realizar un trabajo de tal calibre cuando tienes más responsabilidades que suplir. Además de añadir de que si se añade una hora de ida y otra de vuelta, más media hora para poder comer (10 horas y media). Añadiendo unas ocho horas para poder dormir. El tiempo restante es bastante mínimo para poder llevar a cabo este proyecto.

- **Estar (o no) a la altura de las expectativas.** Cabe decir que mi esfuerzo siempre se ha enfocado para estar a la altura de expectativas y no desilusionar a las personas que van a ver este trabajo. Empezando por mi tutor (del cual su opinión es muy importante para mi) y la valoración del jurado.

9.2. Conclusiones

Después de ver los baches que se han encontrado por el camino siempre se extrae un pequeño aprendizaje de ellos y, por ello, esta sección.

Las conclusiones que se extraen de forma técnica son las siguientes:

- **Objetivos alcanzados. Se ha implementado** un "Sistema de gamificación para la gestión de proyectos software". Es **un sistema** en el que puedan interactuar **distintos actores** (profesor y estudiantes) **en un mismo entorno**. Esto ha llevado a tener que tomar una serie de requisitos enfocado a este objetivo.

- **La interfaz de usuario se ha pensado para que sea lo más intuitiva y fácil de entender desde un primer vistazo.** Además se ha pensado para poder ser utilizado desde smartphones y tablets con un responsive-design. Este diseño hace que se desarrolle una vez pero valga para tres dispositivos distintos:

ordenador, tablet y smartphones. Además se ha realizado una división en los distintos módulos que lo componen para tener una visión más clara.

- La lógica de negocio implementada ha estado enfocada a poder tener una eficiente gestión en la gamificación de la gestión de los proyectos. Esto quiere decir, se puede añadir, editar y borrar los distintos elementos que lo ponen.

- La implementación de la base de datos ayuda a entender mucho mejor el sistema y cómo se comunican los distintos elementos entre ellos. La realización de las consultas es muy sencilla ya que Laravel [4] te permite asociar cada tabla a un "modelo" y eso habilita poder lanzar consultas de una forma rápida.

Me gustaría incluir una serie de conclusiones de ámbito personal y que quizás puedan ayudar a otras personas que se hayan encontrado en mi situación:

- **La realización de un proyecto depende de tus conocimientos.** El enfoque que un proyecto puede tener depende de uno mismo. Si se encuentra en un equipo, esto será un problema inicial que afectará a todo el proyecto.

- **La motivación y tu forma de ser es determinante.** Realizar un proyecto que te motive, que te mueva por dentro y que lo disfrutes no se puede considerar un trabajo. Las personas desmotivadas trabajan peor (y, lo peor de todo, desmotivan al resto de compañeros). También lo es la forma de ser. ¿Cómo asumes el proyecto? ¿Un reto o un fastidio? ¿Crees que es una experiencia fantástica de aprender? Dependiendo de la respuesta a estas sencillas preguntas se puede conocer cuán una persona está predispuesta a realizar un proyecto (¡y que no lo abandone en el camino!).

- **Todo (o casi todo) es posible con tiempo.** Todos los seres humanos por naturaleza somos torpes. Siempre hay excepciones, aunque, por suerte o por desgracia, no me incluyo entre ellos. Especializarte en un tema se tarda tiempo y es una curva de aprendizaje (que posiblemente dure toda una vida conocerla al 100%). El problema es que muchas veces existe el deseo de quererlo de forma muy rápida. En este momento llegan los problemas y las complicaciones.

- **Como conclusión final: Somos nuestro peor enemigo.** Esta lucha interna por intentar mejorar y nuestro cuerpo prefiere descansar. Disciplina y constancia hacen que todo vaya a buenos puertos. Aunque como siempre se dice es más fácil la teoría que la práctica.

Las conclusiones en el resto de ámbitos son las siguientes:

- **En todo proyecto deben plantearse objetivos realistas y que se pueda saber si se han completado o no de forma visible.** En mi opinión es una de las mejores formas de ver si la evolución de un proyecto es correcta. Te dará muchas pistas sobre si has utilizado mucho tiempo, poco o era lo planificado. Una de los hechos que uno aprende es dar importancia al tiempo de uno mismo.

- **Lo óptimo es enemigo de lo bueno.** Mientras más bueno o más funcionalidades tenga, seguramente sea una aplicación poco óptima. Hay que intentar llegar a un término medio ya que nunca se puede llegar a los dos extremos (a la vez).

- **Aprende de otras personas que ya han recorrido el mismo camino o parecido al tuyo.** ¿Otra persona se ha enfrentado a un proyecto parecido al tuyo? Una de las facilidades que da internet es que puedes conocer a personas de todo el globo terráqueo. Posiblemente hayan pensado en hacer algo parecido a lo que planteas. Mira su funcionamiento, cómo lo han ideado y coge lo mejor de cada proyecto. Pensar que uno mismo tiene la solución a todos los problemas es un grave error.

- **No se puede simular todo.** Algo en lo que hay que darse cuenta es que no se puede simular todo lo que ocurre alrededor dentro de una gestión de proyectos. Hay muchas variables y todas ellas distintas entre sí. Lo mejor que se puede hacer es cubrir todos los casos más comunes. Los casos más excepcionales se dan de forma rara y ocasional, por lo que no sería necesario tener que dedicar tiempo a ello.

- **Preparación es un paso más.** Utilizar el simulador para ver cómo sería una gestión de un proyecto es una genial idea ya que no se pierde dinero, se aprenden conceptos y se ven qué planteamientos tenemos de forma correcta en nuestra cabeza (y, sobre todo, cuáles no). Después de prepararse y entrenar, si la idea de uno es saber más de dirección de proyectos software debe evaluar si estudiar un master orientado a la dirección, enfocar su orientación laboral o emprender (teniendo en cuenta todo lo que esto conlleva).

9.3. Futuras mejoras

Es importante conocer y saber cómo se podría mejorar este Trabajo de Fin de Grado. He reunido un listado de qué funcionalidades y mejoras podrían ser añadidas para que fuera un producto más completo y de más calidad.

- **Gestión automática.** Actualmente el administrador o encargado del servicio se encarga de crear, modificar y borrar alumnos. Gestionar productos de la tienda, variación de precios dependiendo del stock, crear y editar eventos. Todo ello se podría realizar de una forma automática con lo que eliminaría una gran cantidad de tiempo perdido.

- **Clasificaciones por equipo.** Una forma de poder visualizar cómo lo realizan el resto de compañeros de clase (si se hiciera para una clase específica) y poder competir entre ellos viendo cómo van evolucionando.

- **Reportes.** Sería interesante mostrar qué decisiones han tomado cada equipo y cuáles han beneficiado más y cuáles han hecho perder. También es interesante para evaluar qué ventaja sobre cualquier elemento del

simulador ha sido la más crítica. Este reporte ayudaría a evaluar qué mejoras se podrían realizar para años consecutivos.

- **Notificaciones.** Las notificaciones son usadas para mandar recordatorios, alertas o notificar de cualquier evento ocurrido. Además de que incrementan el retorno del usuario y le ayudan avisándole de que tiene que tomar una decisión o revisar cualquier asunto pendiente.

- **Encuesta final.** Una forma de conocer la opinión de la persona que ha usado el simulador sería realizar una encuesta después de haber usado la herramienta para poder evaluar qué funcionalidades ayudan, cuáles dan problemas y cuáles se podrían mejorar. El feedback es una fuente valiosa de mejora en cualquier tipo de proyecto, ya que tener al cliente satisfecho (en este caso el usuario del simulador) es un factor diferencial a otro producto similar (si lo hubiera).

- **Aplicación multi-idioma.** Una forma de hacer crecer el servicio es poder ofrecerlo en distintos idiomas. Esto ayuda a que el público objetivo sea mayor y se le pueda ofrecer a más gente.

- **Aplicación móvil.** Uno de los dispositivos que más usamos es el móvil y por ello, no cabría duda que una aplicación móvil sería una mejora ideal. Los dos mercados de aplicaciones móviles más usados son Apple y Google (iOS y Android, respectivamente).

Bibliografía

[1] Ingeniería del software. Un enfoque práctico. Roger S. Pressman

[2] Sharkworld. Último acceso: 20/09/2017. http://www.sharkworldgame.com/

[3] BDD in Action: Behavior-driven development for the whole software lifecycle (2014). John Ferguson Smart.

[4] Laravel: Up and Running: A Framework for Building Modern PHP Apps (2016). Matt Stauffer.

[5] Toms Planner. Último accesso: 20/09/2017 https://www.tomsplanner.es

[6] The Project Management Game. Último acceso: 20/09/2017. http://thatpmgame.com/

[7] Jenkins. Continuous Integration. Último acceso: 20/09/2017. https://jenkins.io/

[#] Software Configuration Management: A How To Guide for Project Staff (2011). Dr David Tuffley.

[#] Managing Software Engineering Knowledge (2003). Aurum, A., Jeffery, R., Wohlin, C., Handzic, M.